Supplementary Material of SC-wLS: Towards Interpretable Feed-forward Camera Re-localization

Xin Wu[1,2⋆], Hao Zhao[1,3⋆], Shunkai Li[4], Yingdian Cao[1,2], and Hongbin Zha[1,2]

[1] Key Laboratory of Machine Perception (MOE), School of EECS, Peking University
[2] PKU-SenseTime Machine Vision Joint Lab
[3] Intel Labs China
[4] Kuaishou Technology
{wuxin1998,zhao-hao,lishunkai,yingdianc}@pku.edu.cn, zha@cis.pku.edu.cn

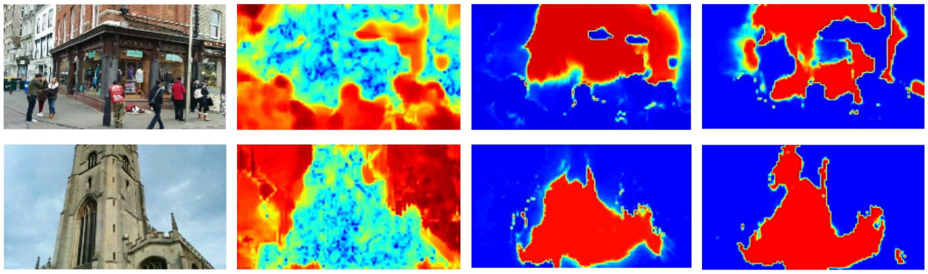
(a) input test image (b) reprojection error (c) weights before fine-tuning (d) weights after fine-tuning

Fig. 1. After self-supervised fine-tuning during test time, the quality network selects better scene coordinates.

1 Evaluations with SfM ground truth on 7Scenes dataset

As illustrated in [1], the reference algorithm used to create pseudo ground truth has an influence on the performance of a certain family of re-localisation methods. Thus we also train (w/o 3D model) and evaluate our method using the pseudo ground truth (pGT) generated by SfM [1] on 7Scenes dataset. In Table 1, we report the median translational and rotational errors and recalls with pose error below 5cm, 5deg, and the settings are the same as Table 1, 4 in the main paper. Compared with the results on the original 7Scenes dataset, the performance of our methods trained with pGT-SfM improves on all of the scenes, which is consistent with other re-localization methods reported in [1]. Besides, it is shown that the recalls of our method Ours (dlt+$e2e$+ref) under-perform DSAC* (w/ model), and using DSAC*'s exact post-processing can compensate for this gap.

[⋆] equal contribution

Table 1. Results on the 7Scenes dataset [6], with translational and rotational errors measured in m and $^\circ$, and recalls (%) with pose error below $5cm$, 5°. Here notations are the same as the Table 1, 4 in the main paper. Note DSAC* [1] w/ model is trained with 3D model, and all of these methods are trained using pseudo ground truth generated by SfM [1].

Median Errors	Chess	Fire	Heads	Office	Pumpkin	Kitchen	Stairs	
Ours (*dlt*)	0.012/0.39	0.030/0.59	0.024/1.53	0.036/0.65	0.038/0.47	0.039/0.73	0.155/2.85	
Ours (*dlt*+*e*2*e*)	0.011/0.37	0.029/0.58	0.022/1.46	0.029/0.59	0.035/0.43	0.033/0.60	0.093/2.03	
Ours (*dlt*+*e*2*e*+ref)	**0.007/0.20**	**0.010/0.34**	**0.009/0.53**	**0.013/0.28**	**0.017/0.32**	**0.010/0.21**	**0.032/0.81**	
Recalls	Chess	Fire	Heads	Office	Pumpkin	Kitchen	Stairs	Avg
DSAC* [1] w/ model	99.9	98.9	**99.8**	98.1	**99.0**	97.0	**92.0**	**97.8**
Ours (*dlt*+*e*2*e*)	90.1	66.0	60.7	75.6	63.2	64.0	28.8	64.1
Ours (*dlt*+*e*2*e*+ref)	100	86.3	68.9	93.9	80.5	91.1	65.9	83.8
Ours (*dlt*+*e*2*e*+$dsac*$)	**100**	**99.7**	97.9	**99.5**	94.4	**97.7**	84.2	96.0

2 More ablation studies

Graph Attention Layer. As described in Section 3.2 in the main paper, we use Graph Attention Layers in the quality weight network. The quality weight network is based on OANet [8], which is a PointNet-like architecture that takes 2D-3D correspondence pairs as input. We propose to introduce the self-attention mechanism into OANet, replacing the original normalized MLP modules with Graph Attention Layers. This layer builds a fully connected graph upon clusters and conducts global message passing between all nodes. This network architecture enhancement improves the re-localization accurary. As shown in Table 2, using the Graph Attention Layer, the median transition error for Ours (*dlt*) is decreased from 23cm to 18cm on *Stairs*, and 19cm to 15cm on *Shop Facade*. We also report the results of Ours (*dlt*+*e*2*e*) in Table 2, which are consistent with Ours (*dlt*) in most scenes.

Classification loss $\mathcal{L}_c$ in Section 3.3 is a binary cross-entropy function which plays the role of a hard outlier pruner. It allows a more stable convergence during training, and enhances the pose estimation accuracy. $\mathcal{L}_c$ is effective for both indoor and outdoor scenes, and especially useful for chanllenge environments, as shown in Table 2. For instance, it results in a relative error reduction of 77% on *Shop Facade*, whose scene coordinates are very noisy due to dynamic objects and non-Lambertian reflection.

3 Self-supervised adaptation at test time

As illustrated in the main paper, self-supervised adaptation deals with the domain shift problem during test time and significantly enhances the re-localization performance of the DLT setting. As a supplement to Fig. 4 in the main paper, we compare learned weights on Cambridge before and after self-supervised adaptation at test-time, in Fig. 1. It shows that the quality network is capable of selecting better scene coordinates after test-time adaptation.

Table 2. More ablation studies on some scenes of 7Scenes dataset [6] and Cambridge dataset [4]. We validate the effects of Graph Attention Layer and classification loss $\mathcal{L}_c$, and report median translational and rotational errors measured in m and °. *dlt* and *e2e* are evaluated without and with the third end-to-end fine-tuning stage mentioned in Section 3.4 of the main paper. The best results are highlighted.

		Ours (*dlt*)				
Attention Layer	$\mathcal{L}_c$	Fire	Office	Stairs	Shop Facade	Church
✓		0.055/1.06	0.076/1.06	0.251/4.16	0.65/2.5	0.62/1.9
	✓	0.060/1.09	0.068/1.03	0.230/4.00	0.19/1.2	0.50/1.5
✓	✓	**0.051/1.04**	**0.063/0.93**	**0.179/3.61**	**0.15/1.1**	**0.50/1.5**
		Ours (*dlt* + *e2e*)				
	✓	0.057/1.10	0.061/**0.84**	0.163/3.06	0.12/0.7	**0.35**/1.3
✓	✓	**0.048/1.09**	**0.055**/0.86	**0.123/2.80**	**0.11/0.7**	0.39/**1.3**

Table 3. Median errors of the pose estimation on 7Scenes dataset [6] w.r.t. the self-supervised adaptation module. Translational and rotational errors are measured in cm and °. *dlt* is the weighted DLT method, *e2e*, *ref* and *self* denote end-to-end training step, LM-Refine and self-supervised adaptation, respectively. 150k means 150k iterations of fine-tuning. The best results are highlighted.

Methods	Chess	Fire	Heads	Office	Pumpkin	Kitchen	Stairs
Ours (*dlt*+*e2e*+*self* [150k])	2.5/0.71	3.3,0.98	2.6/1.84	4.1/0.81	6.1/1.21	6.2/1.26	6.8/1.24
Ours (*dlt*+*e2e*+*self* [600k])	2.1/0.64	**2.3/0.80**	1.3/0.79	3.7/0.76	**4.8**/1.06	**5.1/1.08**	5.5/1.48
Ours (*dlt*+*e2e*+*self*+*ref* [600k])	**1.9/0.62**	2.4/0.86	**1.3/0.69**	**3.4/0.76**	4.9/**1.04**	5.2/1.12	**4.7/1.21**

In the main paper, we report the results of fine-tuning for 600k iterations on 7Scenes and Cambridge datasets. It is important to investigate how well the network adapt with fewer iterations, so as to be practical in real-world applications. As shown in Fig. 2, the median translation error of the estimated poses drops rapidly within 20k iterations (wall-clock time: 23 minutes). Thus, we show additional pose estimation results within fewer iterations (150k, wall-clock time: 2.9 hours), in Table 3 and 4. It demonstrates that the proposed self-supervised adaptation module can achieve reasonably good results using few iterations, and fine-tuning with more iterations proceeds to improve the pose estimation.

Besides, we describe the mechanism of self-supervised adaptation in detail, using Fig. 3. Specifically, we select two adjacent frames as source and target images, and warp I_s to target I_t using the predicted scene coordinates $\mathbf{C_s}$ in

Table 4. Median errors of the pose estimation on Cambridge dataset [4] w.r.t. the self-supervised adaptation module. Translational and rotational errors are measured in m and °. The notations are the same as Table 3. The best results are highlighted.

Methods	Greatcourt	King's College	Shop Facade	Old Hospital	Church
Ours (*dlt*+*e2e*+*self* [150k])	0.95/0.5	0.11/0.4	0.05/0.4	0.20/0.7	0.22/0.9
Ours (*dlt*+*e2e*+*self* [600k])	0.94/0.5	0.11/0.3	0.05/0.4	0.18/0.7	0.17/0.8
Ours (*dlt*+*e2e*+*self*+*ref* [600k])	**0.28/0.2**	**0.08/0.2**	**0.04/0.3**	**0.11/0.4**	**0.09/0.3**

the world coordinate system and the transform matrix $\mathbf{T_{t2w}}$ calculated by the DLT process. Then the photometric error serves as self-supervision and backpropagates gradients along the red arrows, while others are detached in this stage. Note that the $\mathbf{T_{t2w}}$ is post-processed by the algorithm of Section 7, which is fully differentiable and enables the gradients flowing from the photometric loss. In our experiments, the sampling interval of two images is 7 for 7-Scenes and 1 for Cambridge, respectively.

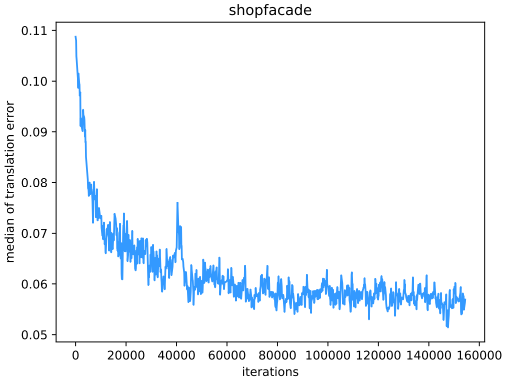

Fig. 2. The median translation error of the estimated poses on *shopfacade* (from Cambridge) during self-supervised adaptation. It rapidly converges within 20k iterations, which demonstrates the effectiveness of this module in real world scenarios.

4 3D Map visualization

In Fig. 5, we visualize point clouds predicted directly by scene coordinate regression networks and those filtered by the learned weights of our SC-wLS framework. We filter out points with weights smaller than $\lambda = 0.9$. It shows that the scene coordinates are particularly noisy and due to the nature of reprojection supervision, point clouds may drift away far along the bearing vectors. However, with the assistance of our SC-wLS, the weight-filtered point clouds are much more accurate and cleaner, showing the effectiveness and interpretability of the learned weights as well.

5 Loss functions

This section elaborates on the hyper-parameter tuning of the regression loss $\mathcal{L}_r$ in Section 3.3. As illustrated in the main paper, we use an eigen-decomposition

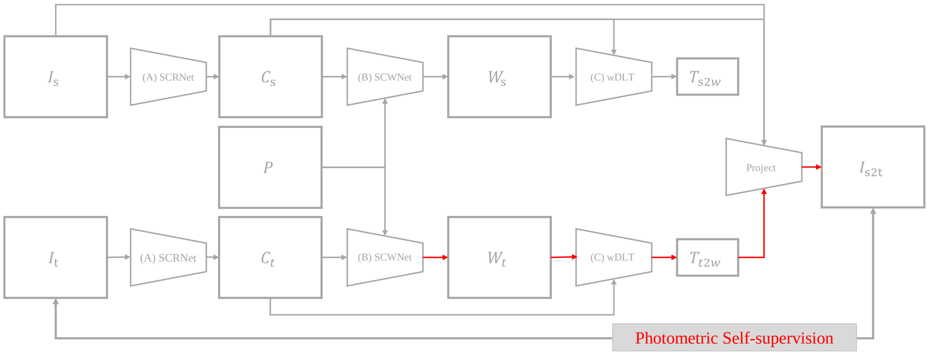

Fig. 3. A detailed illustration of the self-supervised fine-tuning procedure. I_s is the source RGB image. I_t is the target RGB image. SCRNet (A) is the scene coordinate regression network. C_s is the scene coordinate map for the source image. C_t is the scene coordinate map for the target image. P is the 2D pixel coordinate map. SCWNet (B) is the scene coordinate quality weight network. W_s is the scene coordinate weight map for the source image. W_t is the scene coordinate weight map for the target image. wDLT (C) is the weighted least squares solver. T_{s2w} is the camera pose of the source image in the world frame. T_{t2w} is the camera pose of the target image in the world frame. Only tensors that flow through red arrows receive supervision signals while others are detached.

free loss [3] to refrain from the numerical instability caused by the eigen-vector switching problem:

$$\mathcal{L}_r = \mathbf{t}^\top \mathbf{X}^\top \mathrm{diag}(\mathbf{w}) \mathbf{X}\mathbf{t} \ + \alpha \mathrm{e}^{-\beta \mathrm{tr}(\bar{\mathbf{X}}^\top \mathrm{diag}(\mathbf{w}) \bar{\mathbf{X}})} \tag{1}$$

where $\mathbf{t}$ is the flattened ground-truth pose, $\bar{\mathbf{X}}$=$\mathbf{X}(\mathbf{I} - \mathbf{t}\mathbf{t}^\top)$, while α and β are positive scalars.

As mentioned before, those two terms in Eq. 1 serve as different roles. The former originates from $\mathrm{diag}(\sqrt{\mathbf{w}})\mathbf{X}\mathbf{t} \ = 0$, and minimizing this term leads to the trivial solution of $\mathbf{w} = \mathbf{0}$ as well. Thus the latter term is proposed to alleviate its impact. Since $\bar{\mathbf{X}}$ projects all data vectors onto the hyperplane normal to $\mathbf{t}$, we could maximize the trace of $\bar{\mathbf{X}}^\top \mathrm{diag}(\mathbf{w})\bar{\mathbf{X}}$ to make the eigenvalues corresponding to directions orthogonal to $\mathbf{t}$ to be as large as possible. It's important to select proper hyperparameters α and β to balance these two terms. At the same time, the magnitude of the last term varies considerably on different scenes. We recommend to choose β as the inverse of the magnitude of this trace, and in our experiments, α and β are set to 5 and 1e-4 for indoor 7Scenes while 5 and 1e-6 for outdoor Cambridge, respectively.

6 Network architecture

As mentioned in Section 4.1 in the main paper, we adopt the scene coordinate regression network architecture from [2] for 7Scenes. For Cambridge, we use the same architecture as the feature encoder in RAFT [7] to replace the early layers of this network, which has residual connections, as shown in Fig 4. It increases the receptive field size of the network, and enhances the robustness of feature extraction for complicated environments. We have also tried this enhanced architecture for 7Scenes, which does not bring performance improvements.

7 Post-processing of the DLT algorithm

As described in Section 3.1 of the main paper, pose estimation is solved by the Direct Linear Transform (DLT) algorithm:

$$\mathbf{X}^\top \mathrm{diag}(\mathbf{w})\mathbf{X}\mathrm{Vec}(\mathbf{T}) = 0 \tag{2}$$

where the transform matrix is flattened as $\mathrm{Vec}(\mathbf{T})$, and corresponds to the smallest eigen-vector of $\mathbf{X}^\top \mathrm{diag}(\mathbf{w})\mathbf{X}$. To guarantee that the rotation matrix $\mathbf{R}$ is orthogonal and has determinant 1, we post-process the DLT results using the common generalized Procrustes algorithm [5]. The pseudo-code of our implementation is:

Algorithm 1 Post-processing of the pose calculated by the DLT algorithm

```
Input:$\bar{\mathbf{T}} = \begin{bmatrix} p_1 & p_2 & p_3 & p_4 \\ p_5 & p_6 & p_7 & p_8 \\ p_9 & p_{10} & p_{11} & p_{12} \end{bmatrix} = [\bar{\mathbf{R}}_{3\times3}|\bar{\mathbf{t}}_{3\times1}]$, and learned weight $\mathbf{w}$;
Output: Regularized $\mathbf{T} = [\mathbf{R}_{3\times3}|\mathbf{t}_{3\times1}]$;
$\mathbf{U\Sigma V} = \mathrm{SVD}(\bar{\mathbf{R}})$;
$s = \frac{3}{tr(\mathbf{\Sigma})}$;
Selecting the most confident 2D-3D correspondence $c = (u, v, x, y, z)$ according to
the learned weight $\mathbf{w}$;
if $s(xp_9 + yp_{10} + zp_{11} + p_{12}) > 0$ then
    $s = s$;
else
    $s = -s$;
end if
$\mathbf{R} = sign(s)\mathbf{UV}^\top$;
$\mathbf{t} = s\bar{\mathbf{t}}$;
```

It's worth noting that these steps are fully differentiable, thus the self-supervised adaptation in Section 3.6 is able to back-propagate gradients through this operation.

8 Error Metrics

Median errors are robust to outlier estimates. We report the translation and rotation errors per testing frame on *office* and *kingscollege* in Fig. 6. It's shown

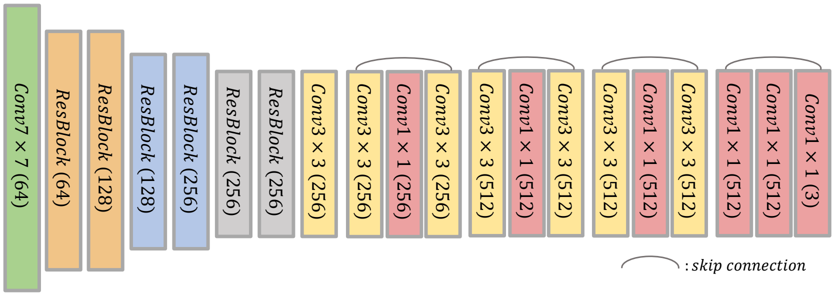

Fig. 4. Scene coordinate regression network details for Cambridge. The first 8 blocks are the same as the feature encoder in [7], and no normalization is used. Skip connections are used in both ResBlocks and the last 12 blocks as denoted.

that per frame error may be negatively impacted by outliers thus hard to tell the true algorithm performances, while median errors are easier to compare between lots of methods. We have released the code for per frame error analysis.

9 Delving into Interpretability

As for interpretability, we report pearson correlation coefficients between learned weights and inverse reprojection errors in Table 5, in which outdoor scenes give higher correlation values due to many uncertain regions like sky and human.

Table 5. Pearson correlation coefficients. ($p < 0.01$)

Chess	Fire	Heads	Office	Pumpkin	Kitchen
0.18	0.15	0.18	0.15	0.21	0.14
Stairs	Court	College	Shop	Hospital	Church
0.21	0.23	0.22	0.23	0.24	0.21

References

1. Brachmann, E., Humenberger, M., Rother, C., Sattler, T.: On the limits of pseudo ground truth in visual camera re-localisation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 6218–6228 (2021) 1, 2
2. Brachmann, E., Rother, C.: Visual camera re-localization from rgb and rgb-d images using dsac. IEEE Transactions on Pattern Analysis and Machine Intelligence (2021) 5
3. Dang, Z., Yi, K.M., Hu, Y., Wang, F., Fua, P., Salzmann, M.: Eigendecomposition-free training of deep networks for linear least-square problems. TPAMI (2020) 5

4. Kendall, A., Grimes, M., Cipolla, R.: Posenet: A convolutional network for real-time 6-dof camera relocalization. In: ICCV (2015) 3
5. Schönemann, P.H.: A generalized solution of the orthogonal procrustes problem. Psychometrika **31**(1), 1–10 (1966) 6
6. Shotton, J., Glocker, B., Zach, C., Izadi, S., Criminisi, A., Fitzgibbon, A.: Scene coordinate regression forests for camera relocalization in rgb-d images. In: CVPR (2013) 2, 3
7. Teed, Z., Deng, J.: Raft: Recurrent all-pairs field transforms for optical flow. In: European conference on computer vision. pp. 402–419. Springer (2020) 6, 7
8. Zhang, J., Sun, D., Luo, Z., Yao, A., Zhou, L., Shen, T., Chen, Y., Quan, L., Liao, H.: Learning two-view correspondences and geometry using order-aware network. In: ICCV (2019) 2

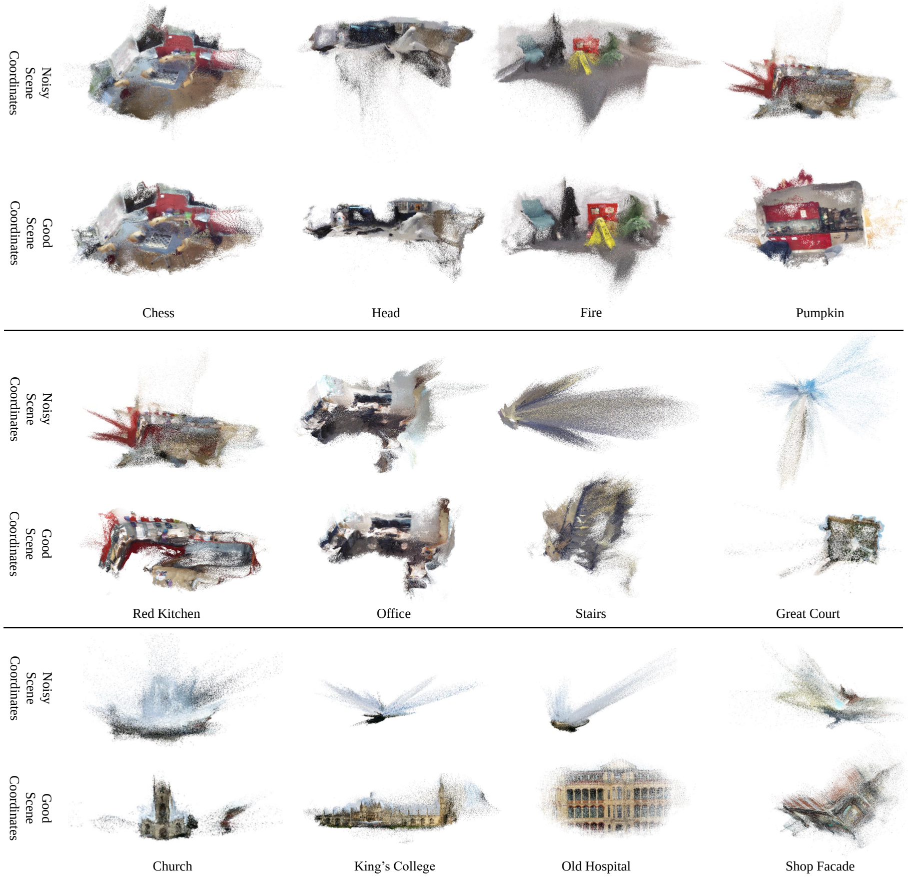

Fig. 5. Map visualization on the test sets of 7-Scenes and Cambridge. Since scene coordinates are predicted in the world frame, we directly show the point clouds generated by aggregating scene coordinate predictions on test frames. It can be seen that only predicting scene coordinates results in noisy point clouds, especially in outdoor scenes where scene coordinate predictions on sky regions are only meaningful in term of their 2D projections. We show good scene coordinates by filtering out samples with a quality weight lower than 0.9.

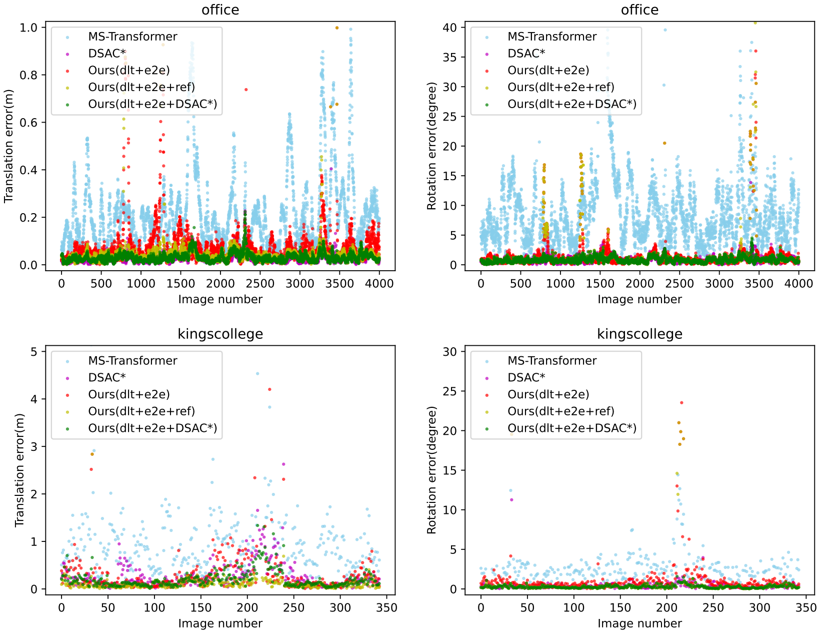

Fig. 6. Translation and rotation errors per testing frame.