A. Discussion

A.1. Difference from matching methods

Our work focuses on the detection and description of key-points, without a learnable matching strategy. We only compare our method with other local features (including hand-crafted and learning-based methods) for a fair evaluation during the experiments. All the matching results in Table 1, Table 3, and Table 4 are based on the **built-in Nearest-Neighboring matcher**, instead of additional learnable matching model [42, 62] or end-to-end matching method [20, 48]. Furthermore, our learning transformation-predictive representations can be used in various local features from CNN- to Transformer-based. Finally, the learned local features can be taken as the input of matching methods to produce better image matching and visual localization results like HLoc [11, 42].

A.2. Self-supervised learning in local features

Recently, a surge of interest has emerged in self-supervised contrastive learning for deep neural networks. Based on the self-supervised pre-trained backbone, the fine-tuned or linear-projected models show competitive or better performance on downstream computer vision tasks (*e.g.*, classification, detection, and segmentation). While our self-supervised TPR is trained with single-stage joint learning, different from the above two-stage cascaded learning methods.

On the other hand, learning local features is a problem that cannot be tackled by standard supervised training, as observed in previous works [11, 60]. Furthermore, the local features are ill-defined and are hard to manually annotate. We thus treat the training local features as a self-supervised task with the detection and description loss function. Our TPR uses affine adaption and data augmentation to generate the ground-truth correspondences, supervising the detector and descriptor of local features. Here we report the training data (fewer constraints is better), model size (smaller is better), and the dimension of the descriptor (lower is better) of learning-based methods in Table 5. Our method requires no SfM data or extra information, which is self-supervised only on the images from the web, which are generally easy to collect and scale up.

A.3. Difference from soft labels in Teacher/Student Networks

Previous methods set all the similarities of positive pairs as "1", which can be considered as the inductive bias of the contrastive learning models. Our TPR method trains the local features with only positive pairs. Therefore, we soften the hard label "1" into the soft objectives for different positive pairs with different transformation scale. Compared with using one-hot embedding to train the model, soft labels have been widely adopted in knowledge distillation (KD), *i.e.*, Teacher/Student Networks. The soft labels produced by the pre-trained Teacher model, are different from our soft labels predicted from curriculum learning and self-supervised generation learning in §3.3. Our joint learning process is single-stage, requiring none of the pre-trained Teacher models. Furthermore, our soft labels are predicted to fix the **label noise** caused by the false positive pairs with over-strong transformation. While the one-hot embedding in KD is usually human-labeled, which is the correct label.

Method	Training Data	Model(MB)	Dim
D2-Net [12]	SfM data	30.5	512
DELF [35]	landmarks data	36.4	1024
LF-Net [36]	SfM data	31.7	256
SuperPoint [11]	synthetic&web images	5.2	256
R2D2 [40]	web images, SfM data	2.0	128
Disk [52]	SfM data	1.1	128
Ours(VGG)	web images	30.5	512
Ours(TR)	web images	57.3	128

Table 5. The training data of learning-based methods.

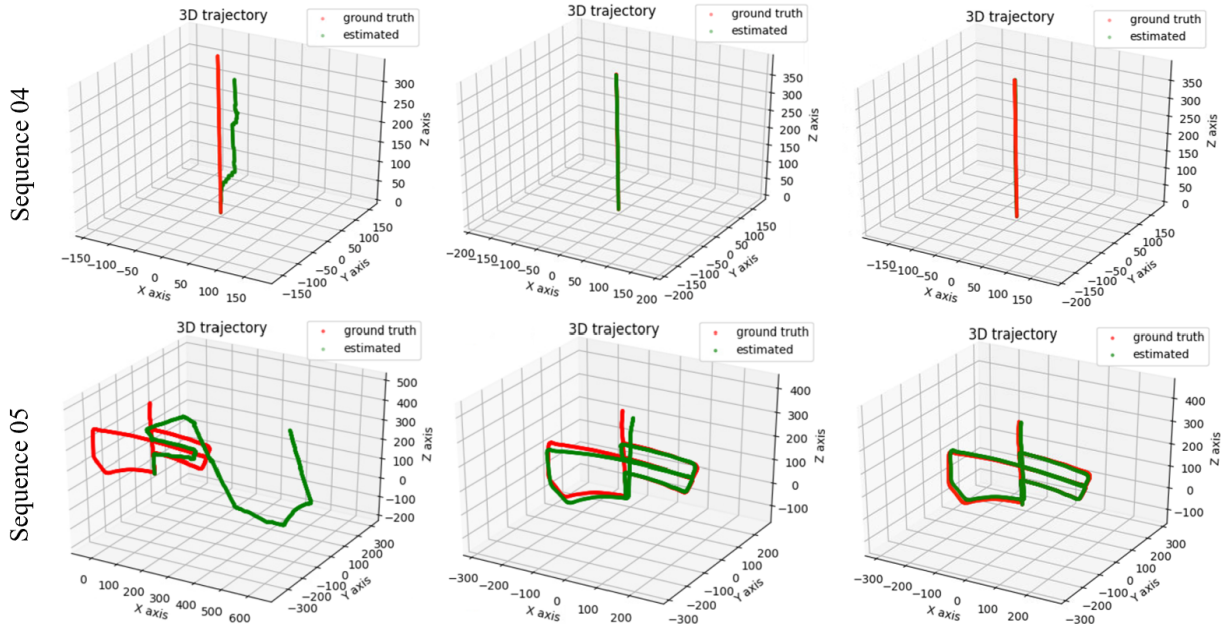
B. Experimental results

B.1. Trajectory visualization in visual odometry

We report the results of the Visual Odometry localization on KITTI dataset in Table 3 and trajectory visualization results in Figure 4. The sequences 04 and 05 have different motion range as (0.5×394)m and (479×426)m, different length of sequences as 272 and 2762, respectively. In sequence 04, the camera's motion trajectory is basically a straight line. With relatively simple motion trajectories (sequence 04), both SuperPoint [11] and our method based on Transformer show good localization performance. However, in the more complex trajectory (sequence 05), our method significantly outperforms the previous method in terms of localization error. In addition, our method achieves the best performance in almost all sequences (00-10) according to RMSE in Table 3. We also report the running time in Table 3, our method does not introduce significant computational complexity compared to previous CNN methods, which can run real-timely.

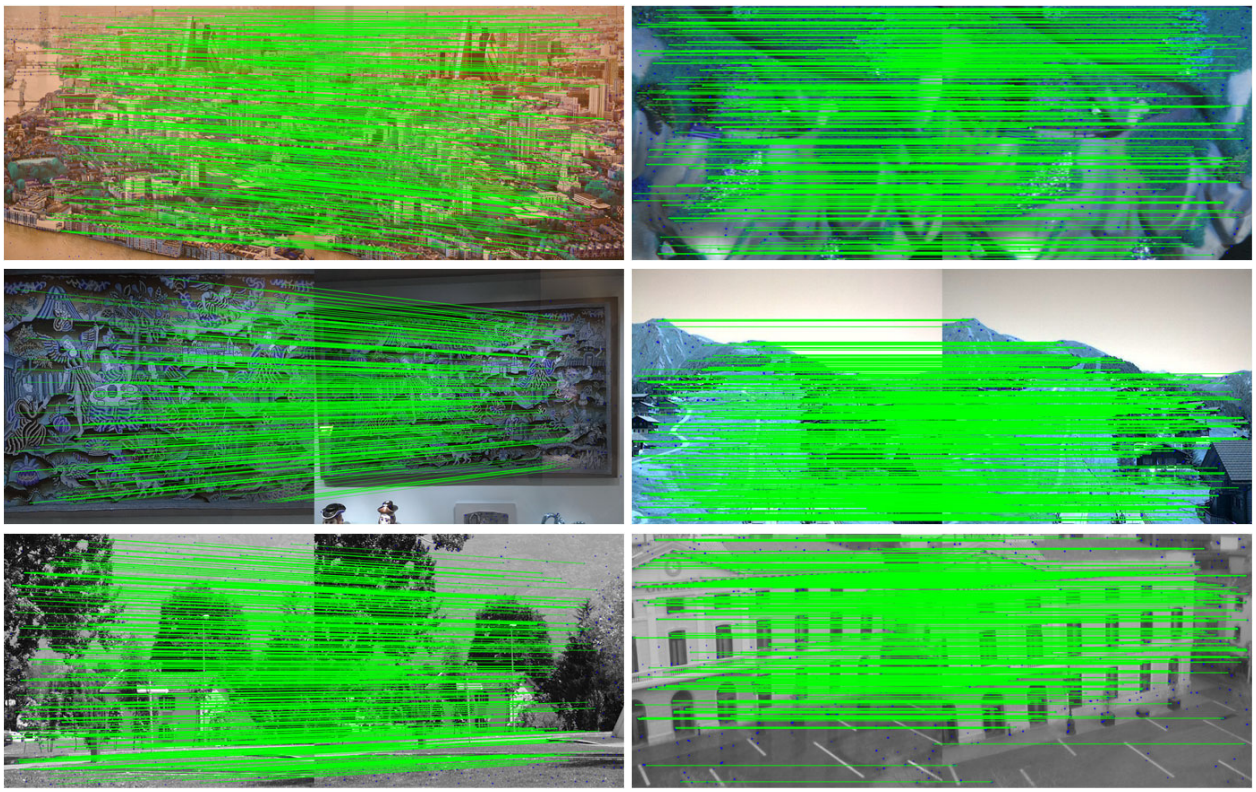
B.2. Image matching results

We also report the visualization matching results on the HPatches dataset [2] in Figure 5. The local features are generated by the networks with swin transformer [27] blocks, which are trained with our learning transformation-predictive representation methods. The matching results are also based on the built-in nearest-neighboring matcher.



(a) ORB (b) SuperPoint (c) Ours

Figure 4. **Visualization of 3D trajectory localization error of different key-points on KITTI 04 and 05 sequences.** The green and red line represent the estimated and the ground truth trajectory, respectively.



ViewPoint Changes Illumination Changes

Figure 5. **Visualization of matching results on HPatches.**