Appendix

A Implementation Details

In this section, we provide more details about our network implementation.

A.1 Network Settings

We use the same ResNet-18 for initial feature extractor as that in LoFTR, which outputs feature maps in two resolution, $\frac{1}{8}$ and $\frac{1}{2}$. The $\frac{1}{8}$ feature map is passed into our transformer-based network for updating, while the $\frac{1}{2}$ is used in fine matches coordinates refinement. For dual-softmax in coarse matching, we adopt a learnable temperature which is initialized as 10.

We use four GLA blocks to update features. For hierarchical attention, we fix the coarsest feature maps in resolution H_0, W_0, where $(H_0, W_0) = (15, 20)$ for indoor settings and $(H_0, W_0) = (36, 36)$ for outdoor settings.

A.2 Flow Regression

As stated in Sec. 3.4, we use an MLP to regress auxilary flow map in each GLA block. Given D-dimensional feature in pixel, we use MLP with shape (D,64,4) to regress a 4-dimensional feature f. For corresponding coordinates u_x, u_y, We normalize the first two values with sigmoid function and recover them to the range of image resolution. For the standard variance σ_x, σ_y, we regress the last two values as their logarithm. Formally,

$$[u_x, u_y] = Sigmoid(f[:2]) * [H, W], \ [\sigma_x, \sigma_y] = exp(f[2:]) \tag{1}$$

where H, W are image height and width.

A.3 Training Details

For both indoor and outdoor training, we adopt the same muti-step training strategy as that in officially released LoFTR code. More specifically, the learning rate is linearly warmed-up in this first epoch and then halved every two or three epochs. The learning rate curve is illustrated in Fig. 1.

A.4 Visual Localization Details

We refer to hierachical localization pipeline (https://github.com/cvg/Hierarchical-Localization) to perform viusal localization experiments on Aachen Day-Night and InLoc datasets.

For Aachen Day-Night, we first triangulate reference models by using only coarse matches across images. We then generate fine level matches between query

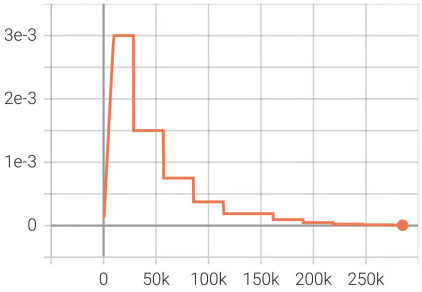

Fig. 1. Learning rate curve across iterations.

images and database images, where the database images are taken as left images, so that the fine level matches can be registered to triangulated 3D tracks.

For InLoc dataset, we directly generate fine level matches between query and database images, where the 2D match points on reference images are projected to 3D space through the provided depth map. We omit image pairs with fewer than 25 matches.

A.5 Some Effective Designs

We provide ablations for some additional useful designs in our network: (1) learnable temperature for softmax at each level. (2)Convolution-based FFN. (3) Normalized positional encoding when testing resolution differs from training resolution. An ablation study for these techniques is provided in Tab. 1 and Tab. 2.

Learnable Temperature. As stated in Sec. 3.4, message M^f, M^m, M^c are computed from different levels of feature maps through global or local attention, where softmax are applied to tokens in different numbers. A concern about softmax is the that the number of tokens largely affect the final distribution. To balance the impact of different token number in global/local attention, we adopt three learnable temperature parameters τ_f, τ_m, τ_c for softmax in fine, medium and coarse level features respectively.

Convolutional FFN. As shown in Sec. 3, our networks is fully based on cross attention for cross-view message passing, while self attention is absent. Deviating from common practice that employs self attention for intra-image message passing, we find in our experiment that adopting 3×3 convolution in FFN to replace self attention and MLP-based FFN leads to better overall performance.

Normalized Positional Encoding. Positional encoding (PE) in LoFTR is defined as,

$$PE^i(x,y) = \begin{cases} sin(w_k \cdot x), & i = 4k \\ cos(w_k \cdot x), & i = 4k+1 \\ sin(w_k \cdot x), & i = 4k+2 \\ cos(w_k \cdot x), & i = 4k+3 \end{cases}$$

A concern about this PE is that unseen coordinate will be used in encoding when testing resolution differs from training resolution, which harms the network's capability of precise localization and boundary awareness. To mitigate the issue, we adopt a simple normalization technique,

$$PE^i_n(x,y) = PE^i(x * \alpha, y * \beta) \tag{2}$$

$$\alpha = W_{tain}/W_{test}, \ \beta = H_{train}/H_{test} \tag{3}$$

where $W/H_{train/test}$ are width/height of training/testing image. We find this normalization boost the performance of our method when training/testing image resolution differ. Aligning testing/training PE is especially critical for precise flow prediction, since it relies on PE to regress flow coordinate.

In Tab. 2, we provide ablation study results for normalized positional encoding (NPE). The results are obtained on MegaDepth dataset with all images resized to 1152 resolution, while the models are trained in 832 resolution.

Table 1. Ablations on network designs on ScanNet [1] dataset. SA+MLP-FFN, means adopting 1/4 downsampled self attention after each GLA block and replacing all 3×3 conv in FFN of both self/cross attention with MLP.

Method	Pose Estimation AUC		
	@5°	@10°	@20°
AspanFormer w/o learnable temperature	25.0	45.7	62.3
AspanFormer w SA+MLP-FFN	24.8	45.5	62.0
AspanFormer	**25.6**	**46.0**	**63.3**

Table 2. Ablation study of Normalized Positional Encoding (NPE) on MegaDepth dataset [2].

Method	Pose Estimation AUC			Flow Acc.
	@5°	@10°	@20°	
AspanFormer w/o NPE	52.8	69.6	81.1	22.6
AspanFormer	55.3	71.5	83.1	72.3

B Flow Loss

We formulate flow supervision as most-likelihood estimation for Gaussian distribution P.

$$L_{flow} = -\frac{1}{|D^{gt}|} \sum_{ij} log(P(D^{gt}_{ij}|\Phi_{ij})) \tag{4}$$

where $D_{ij}^{gt} = (x_{ij}, y_{ij})$ is the ground truth flow and $\phi_{ij} = (u_x^{ij}, u_y^{ij}, \sigma_x^{ij}, \sigma_y^{ij})$ are predicted parameters at location (i, j). Substituting into Gaussian distribution formula, we have

$$L_{flow} = -\frac{1}{|D^{gt}|}\sum_{ij} log[\frac{1}{2\pi\sigma_x^{ij}\sigma_y^{ij}}\exp(-\frac{(x_{ij}-u_x^{ij})^2}{2\sigma_x^{ij\,2}} - \frac{(y_{ij}-u_y^{ij})^2}{2\sigma_y^{ij\,2}})] \tag{5}$$

$$= \frac{1}{|D^{gt}|}\sum_{ij}[log2\pi + log\sigma_x^{ij} + log\sigma_y^{ij} + \frac{(x_{ij}-u_x^{ij})^2}{2\sigma_x^{ij\,2}} + \frac{(y_{ij}-u_y^{ij})^2}{2\sigma_y^{ij\,2}}] \tag{6}$$

In implementation, we let $w_x^{ij} = log\sigma_x^{ij}, w_y^{ij} = log\sigma_y^{ij}$ and omit constant terms, then

$$L_{flow} = \frac{1}{|D^{gt}|}\sum_{ij}[w_x^{ij} + w_y^{ij} + \frac{1}{2}e^{-2w_x^{ij}}(x_{ij}-u_x^{ij})^2 + \frac{1}{2}e^{-2w_y^{ij}}(y_{ij}-u_y^{ij})^2] \tag{7}$$

Intuitively, this loss formulation is a weighted sum of L2-distance between estimated flows and ground truth flows. $w_x^{ij}+w_y^{ij}$ is a regularization term encouraging lower uncertainty. The overall effect of flow loss is to minimize uncertainty and flow estimation error simultaneously.

C Additional Quantitative Results

We provide in this part additional experiment results on YFCC100M dataset and Image Matching Challenge 2022 (IMC 2022) kaggle benchmark.

C.1 Results on YFCC100M

YFCC100M contains a collection of internet images across various tourism landmarks. We adopt the test set from 4 selected landmark sequences as is done in previous works [3–5]. 1000 image pairs are sampled from each sequence, which yields 4000 pairs test set in total. We use OpenCV ransac for two-view pose estimation, where the RANSAC threshold for **all methods** is set to 5×10^{-4} in normalized image coordinate space. Experiment results are given in Tab. 3, where our method outperforms all comparative methods.

C.2 Results on Image Matching Challenge 2022

We submit our method to Image Matching Challenge (IMC) 2022 and report the results in Tab. 4. We resize the input image to a fixed resolution [1472,832] and use OpenCV USAC_MAGSAC to estimate fundamental matrix, where the RANSAC threshold is set to 0.2 pixel. The results show that our method consistently outperforms other strong comparative baselines.

Table 3. Two-view pose estimation results on YFCC100M [6] dataset in outdoor scenes.

Method	Pose Estimation AUC		
	@5°	@10°	@20°
SP [7]+SuperGlue [4]	38.1	58.8	74.7
RootSIFT+SGMNet [5]	35.5	55.2	71.9
DRC-Net [8]	29.5	50.1	66.8
PDC-Net+(H) [9]	39.1	60.1	76.5
LoFTR [10]	42.4	62.5	77.3
Ours	**44.5**	**63.8**	**78.4**

Table 4. Two-view pose estimation results on IMC 2022 kaggle benchmark. The Results of MatchFormer and QuadTree attention are reported by the 4th solution on Kaggle discussion forum [11].

Method	Pose Estimation mAA	
	Private	Public
SP [7]+SuperGlue [4]	0.724	0.728
LoFTR [10]	0.783	0.772
MatchFormer [12]	0.783	0.774
QuadTree [13]	0.817	0.812
Ours	**0.838**	**0.833**

D Additional Visualizations

We provide more visualization results in this part. In Fig 2, we provide qualitative comparisons between SuperGlue, LoFTR and our methods. In Fig 3, we provide flow predictions across GLA block iterations. In Fig 4, we provide additional visualization of uncertainty heatmap and corresponding adaptive attention spans.

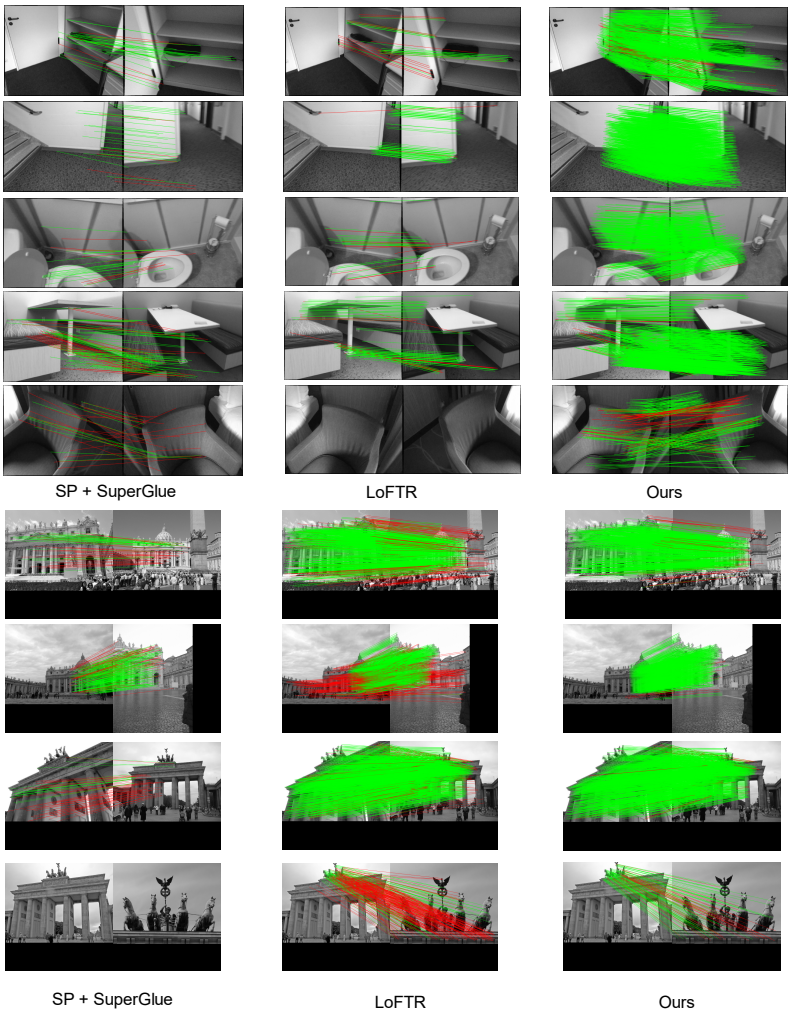

Fig. 2. Visualizations of matches obtained through SuperGlue, LoFTR and ASpan-Former(ours). Our methods produces more accurate and denser matches compared with both SOTA sparse and dense matching networks.

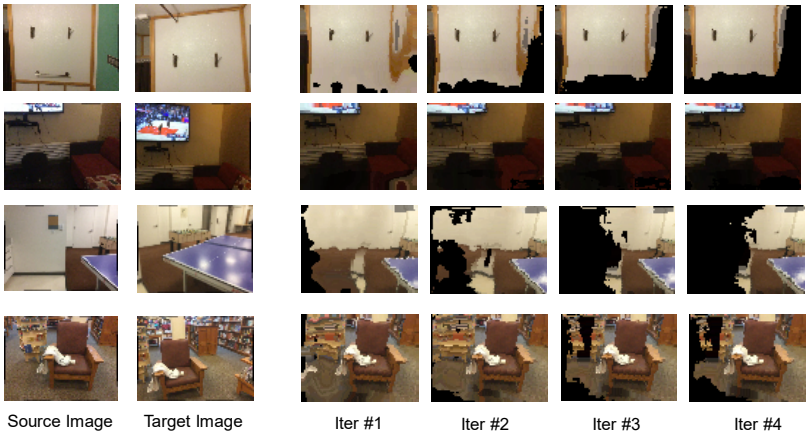

Fig. 3. Visualizations of flow prediction across GLA iterations. We filter flow predictions with high uncertainty. Note that the flow map are in $\frac{1}{8}$ (60×80) resolution. As more GLA blocks are employed for feature updating, the flow map gradually prune occluded or non-overlap regions.

Fig. 4. Visualizations of uncertainty heatmap(left) and adaptive attention span(right). Our network sharply focuses on regions with rich and distinctive textures with small attention span, while larger contexts are extracted for the low texture or uncertain regions. Specially, very large attention spans are generated for non-overlapping or occluded areas, preventing falsely focusing on certain regions.

References

1. Dai, A., Chang, A.X., Savva, M., Halber, M., Funkhouser, T., Nießner, M.: Scannet: Richly-annotated 3d reconstructions of indoor scenes. In: CVPR. (2017)
2. Li, Z., Snavely, N.: Megadepth: Learning single-view depth prediction from internet photos. In: CVPR. (2018)
3. 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)
4. Sarlin, P.E., DeTone, D., Malisiewicz, T., Rabinovich, A.: Superglue: Learning feature matching with graph neural networks. In: CVPR. (2020)
5. Chen, H., Luo, Z., Zhang, J., Zhou, L., Bai, X., Hu, Z., Tai, C.L., Quan, L.: Learning to match features with seeded graph matching network. In: ICCV. (2021)
6. Thomee, B., Shamma, D.A., Friedland, G., Elizalde, B., Ni, K., Poland, D., Borth, D., Li, L.J.: YFCC100M: The new data in multimedia research. Communications of the ACM (2016)
7. DeTone, D., Malisiewicz, T., Rabinovich, A.: Superpoint: Self-supervised interest point detection and description. In: CVPRW. (2018)
8. Li, X., Han, K., Li, S., Prisacariu, V.: Dual-resolution correspondence networks. In: NeurIPS. (2020)
9. Truong, P., Danelljan, M., Timofte, R., Van Gool, L.: PDC-Net+: Enhanced probabilistic dense correspondence network. Preprint (2021)
10. Sun, J., Shen, Z., Wang, Y., Bao, H., Zhou, X.: Loftr: Detector-free local feature matching with transformers. In: CVPR. (2021)
11. Lashkov, I.: 4th solution of imc 2022. https://www.kaggle.com/competitions/image-matching-challenge-2022/discussion/328805
12. Wang, Q., Zhang, J., Yang, K., Peng, K., Stiefelhagen, R.: Matchformer: Interleaving attention in transformers for feature matching. Preprint (2022)
13. Tang, S., Zhang, J., Zhu, S., Tan, P.: Quadtree attention for vision transformers. In: ICLR. (2021)