Extreme Rotation Estimation in the Wild –Supplementary Material–

Hana Bezalel[1] Dotan Ankri[1] Ruojin Cai[2] Hadar Averbach-Elor[1,2]
[1]Tel Aviv University [2]Cornell University

Contents

1. Construction of the *ELP* Dataset 1

2. Implementation Details 1
- 2.1. Network Architecture . . . 1
- 2.2. Input Euler angles . . . 2
- 2.3. Auxillary Channels . . . 2
- 2.4. Training Details . . . 2
- 2.5. Training with Data Augmentations . . . 3
- 2.6. Baselines . . . 4

3. Additional Results and Visualizations 5

4. Additional Ablations 6
- 4.1. The impact of our progressive training scheme 6
- 4.2. Architectural Ablations . . . 8

We refer readers to the interactive visualizations at our project page that show results for all presented models on the *ELP* test sets. In this document, we provide details regarding our proposed dataset (Section 1), additional implementation details (Section 2) and describe additional experiments and results (Section 3).

1. Construction of the *ELP* Dataset

As mentioned in the main paper, we leveraged scene-level Internet photo collections for constructing the *ELP* dataset, focusing on pairs with predominant rotational motion. Given scale is a degree of freedom in SfM reconstruction algorithms, we established a scene-based translation threshold as described below. We construct *mutual* nearest neighbors edge-weighted graphs, with one graph per landmark. In each graph G, nodes $v \in V$ correspond to images, and two images are connected by an edge $e \in E$ if they are both among each other's K nearest neighbors (K is empirically set to 5). The weights w of the edges in each graph G are set according to the L2 distances between the translation of the images. For each landmark, we compute the weighted node degree d_v for each node, defined as the sum of the edge weights of edges incident to that node, divided by the number of such edges.

For example, for an image node v_i with translation T_i and its mutual nearest neighbors represented by nodes v_j, where $j \in [0, ..., k]$ and $k \leq K$, the weight of the edge $e_{i,j}$ is calculated as $w_{i,j} = ||T_i - T_j||_2$. The weighted node degree d_{v_i} is then computed as $d_{v_i} = \frac{1}{k}\sum_{j=0}^{k} w_{i,j}$. Finally, for the entire scene graph, we select image pairs with edge weights w below the median value of weighted node degrees, specifically where $w < \text{med}\{d_v\}$.

The images for *ELP* training set were curated from MegaScenes [14] that utilizes COLMAP [12] for its ground truth poses and uses a Manhattan world alignment. Our benchmark contains two test set, as follows:

***s*ELP**. Image pairs in the *s*ELP test set contain images from the Cambridge Landmarks [1] dataset. The ground truth poses of Cambridge Landmarks Dataset [1] are based on VisualSFM [17]. The ground truth poses were rotated so coordinate system would align with the gravity and horizontal axis using [8].

***w*ELP**. Image pairs in the *w*ELP test set contain images from the MegaDepth [7] dataset. MegaDepth also utilizes COLMAP [12] for its ground truth poses and use Manhattan world alignment.

Both test sets underwent a filtering process to remove any images where transient objects occupied over 40% of the image area. This selection was made using a SegFormer [5] segmentation mask, targeting specific transient object categories. Specifically, we consider: *person*, *car*, *bus*, *bicycle*, *boat*, *truck*, *airplane*, *van*, *ship*, *minibike*, and *animal*.

In Table 1, we provide the image pair distribution across the different scenes of *s*ELP and *w*ELP. The distributions for additional sets are available in the *dataset* directory.

2. Implementation Details

2.1. Network Architecture

Our approach employs an encoder-decoder architecture to predict three Euler angles of relative rotation. Specifically,

Scene name	Scene number	Large	Small	None	All
***s*ELP**					
Total	-	2512	827	1961	5300
GreatCourt	-	548	163	248	959
KingsCollege	-	409	12	0	421
StMarysChurch	-	405	233	35	673
OldHospital	-	395	70	0	465
Street	-	494	342	1678	2514
ShopFacade	-	261	7	0	268
***w*ELP**					
Total	-	2700	829	643	4172
Trafalgar Square, London	1	566	207	157	930
San Marco,Venice	8	226	121	138	485
Piazza del Popolo, Rome	17	350	34	5	389
Vatican, Rome	15	206	101	68	375
Piazza del Campo, Firenze	115	186	111	37	334
Red Square, Moscow	559	231	64	26	321
Piccadilly Circus, London	16	166	92	55	313
Wenceslas Square, Prague	306	237	10	44	291
Washington Square Park, New York City	102	222	4	2	228
Gendarmenmarkt ,Berlin	258	87	11	21	228
Place des Vosges ,Paris	294	53	21	12	86
Grand Place,Brussels	61	21	23	36	80
Royal Mile,Edinburgh	162	52	10	5	67
Bruges	224	33	4	24	61
Grote Markt, Antwerp	472	32	6	10	48
Old Town , Stockholm	238	16	7	2	25
Marienplatz , Munich	65	16	3	1	20

Table 1. The distribution of image pairs across different 3D scenes in the test sets of *ExtremeLandmarkPairs* (*s*ELP and *w*ELP). The scene number denotes the number of landmark in MegaDepth [7] dataset.

we utilize LoFTR as our image encoder, extracting its dense features (after LoFTR's stage 2) with the dimensions of $256 \times \frac{H}{8} \times \frac{W}{8}$. Then, by concatenating the two features along the third dimension, we obtain a feature map of size $256 \times \frac{H}{8} \times \frac{2W}{8}$. We augment the feature dimension by concatenating three auxiliaries masks: keypoints, matches and segmentation mask. This augmented input is then projected into the transformer decoder embedding space, which has a dimensionality of 256. Additionally, we introduce three learnable tokens. Our rotation estimation transformer builds upon the DinoV2 ViT architecture with a patch size of 1, 4 attention heads, and a depth of 8. Finally, the transformer's output is normalized and average-pooled. To estimate the relevant angle, we concatenate the averaged feature token with the corresponding learnable token. Next, we input this concatenated representation into a stack of three fully connected layers resulting in a 360-dimensional output distribution. Overall, our model comprises approximately 80 million parameters in total, including LoFTR and SegFormer, with 22 million of those being learnable parameters.

2.2. Input Euler angles

The Euler angles that are fed into the rotation estimation transformer are learnable tokens (previously demonstrated in [6] and [15]) These tokens are initially set with random values drawn from a standard normal distribution. During training, they adaptively identify and focus on relevant image tokens specific to each Euler angle. During inference, these learnable tokens are initialized by loading from the optimized weights.

2.3. Auxillary Channels

Keypoints & Matches Masks The keypoints are extracted from the LoFTR output. Additionally, for the matches mask, we use geometric verification using RANSAC (reprojection error set to 1 and confidence to 0.99) to estimate the Fundamental matrix and filter out any outliers. Only the keypoints with a confidence value which is greater than 0.8 are considered for geometric verification. We create binary masks for both keypoints and matches, which are then rescaled to match the dimensions of LoFTR features ($\frac{H}{8} \times \frac{W}{8}$). Finally, we concatenate these image masks side by side to obtain the resulting size of $2 \times \frac{H}{8} \times \frac{2W}{8}$.

Segmentation Mask The segmentation map was generated using SegFormer-B3. SegFormer [5] has demonstrated strong performance on outdoor image datasets, such as Cityscapes and ADE-20K. We consider the following categories: *sky*, *building*, *road*, *sidewalk*, *streetlight*. All remaining labels are labeled as *other*. Additionally, *road* and *sidewalk* are grouped together, as the borders of their masks are noisy and both labels have similar 3D spatial context. Finally, we resize the modified segmentation mask to match the LoFTR feature dimensions using NEAREST_EXACT interpolation mode. The resulting concatenated mask has a dimensionality of $1 \times \frac{H}{8} \times \frac{2W}{8}$.

2.4. Training Details

In all of our experiments, we used Adam optimizer ($\beta_1 = 0.5, \beta_2 = 0.9$). The first stages are trained with a single learning rate set to 1×10^{-4}, the *ELP* and ΔIm stages are trained with learning rate of 1×10^{-5}. The batch size is 20, except when finetuning over *ELP*, where it is adjusted to 40 as it achieved a cleaner convergence. The training duration on one Nvidia RTX A5000 are as follows: [5 days, 3 days, 1 hour, 12 hours] for the [[3],ΔFoV,ΔIm,*ELP*] stages, respectively. The total number of epochs for the training process is 34. The number of iterations sufficient to convergence is roughly 700K iteration for the first two stages and 3K for the last 2 stages. While training on the *ELP* dataset, we addressed the imbalance of overall relative rotations for non overlapping pairs in *ExtremeLandmarkPairs* by using a weighted random sampler. This sampler assigned weights based on the overall rotation angle. Additionally, for the overlapping pairs, due to the overlap categories imbalance (40000 images for large overlap and 15492 for small overlap), we employed a weighted random sampler that weighted by the overlap category. We used the balanced validation split of *ExtremeLandmarkPairs* to

monitor training progress of *ELP* (with a stopping criterion of MGE improvement dropping below 0.5°).

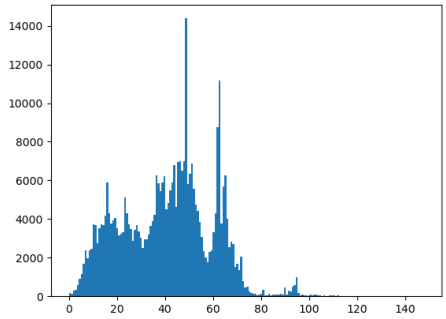

Figure 1. **Histogram of FoVs, corresponding to the *ExtremeLandmarkPairs* training set**. As mentioned in the text, this was used for sampling square perspective images for ΔFoV with a distribution that closely resembles in-the-wild image pairs.

2.5. Training with Data Augmentations

Field of View Augmentations (ΔFoV) To better model the distribution of FoV associated with in-the-wild image pairs, we analyzed the distribution of *ELP* training set, as shown in Figure 1 (considering fov_x and fov_y together). The data revealed that both the median and average FoV values significantly deviate from 90 degrees which is assumed by prior work. Therefore, to better resemble to in-the-wild images, we curated a new dataset of panorama perspective images with a range of FoVs. These images were cropped into squares, with a FoV value selected from a Gaussian distribution with mean set according to the *ELP* training set mean ($\mu_{ELP} = 40.9°$) and a standard deviation which is 1.5 times the standard deviation of the *ELP* training ($\sigma_{ELP} = 21.2$). The distribution was adjusted to exclude values below 30° and above 90° to maintain reasonable image quality. This was achieved by clipping these regions and subsequently renormalizing the distribution. Following DUST3R [16], we also crop the images by the following aspect ratios [(256, 192), (256, 168), (256, 144), (256, 128)]. After cropping, we added zero padding to the remaining areas to achieve a uniform size of 256x256 for all images. For each batch, a single aspect ratio was chosen. We follow the method introduced in [3], first training on overlapping pairs and then on non-overlapping pairs. The overlap training set includes 432992 pairs (35% large overlap, 65% small overlap), non overlap training set includes 1067764 pairs (15% large overlap, 30% small overlap and 55% non overlapping pairs).

Image-Level Appearance Augmentations (ΔIm) As mentioned in the main paper, to perform image-level appearance augmentations, we apply the conditional InstructPix2Pix[2] model on a subset of our data. The InstructPix2Pix editing process uses images that have been augmented with different field of views. The training set for this stage consists 18913 pairs. The model, instruct-pix2pix-00-22000.ckpt, was configured with a text coefficient (cfg-text) of 7.5 and an image coefficient (cfg-image) of 1.5, over a total of 100 steps.

Given that InstructPix2Pix's parameters are applied uniformly across all images, individual responses to the edits can vary, occasionally altering the image's structure. Such alterations could potentially interfere with the cues necessary for estimating relative rotation. To mitigate this, a post-processing filtering stage was implemented in order to remove images whose structure was modified.

The filtering process quantifies structural consistency by analyzing the primary scene boundaries (predominantly the skyline) through a comparative analysis of the segmentation maps from both the source image and its InstructPix2Pix transformation. These segmentation maps are generated using SegFormer[5]. Prior to filtering, we exclude images that lack static structural elements (roads, buildings) or containing predominantly transient features like cars and people. Additionally, indoor scenes (e.g., tunnels) are excluded due to their limited relevant segmentation labels, which generate noisy segmentation maps.

To identify the main borderlines, we focus on the specific labels 'building', 'road', 'sky', 'tree', and 'car', and apply the softmax function exclusively to these categories. We define a main category (C) whose boundary will serve as the major boundary. The main category is selected as the first available mask in the following order: 'sky', 'building', 'road'. The categories not chosen as the main category are defined as secondary categories.

The binary mask M_c is designed to highlight changes in the major boundary and assign a score reflecting the degree of change. Let M_o represent the binary mask of C of the original image and M_t the binary mask of C after InstructPix2Pix transformation. To identify the original boundary, we apply erosion and dilation techniques to M_o (using a disk size of 5 for 'sky' and 10 for 'building' and 'road'). The difference between the eroded and the dilated mask of the original image creates a pronounced border around the main category, denoted as M_b^o. Next, we construct M_c, the binary mask for the altered pixels, using the formula $M_c = M_o \cup M_t - M_o \cap M_t - M_b^o$. Additionally, we exclude any transient elements like 'trees' and 'cars' from the count of altered pixels in M_c.

The score for the major borderline S_c, is then calculated by the following formula $S_c = 1 - \frac{\sum_{i,j} M_{c\,i,j}}{\sum_{i,j} M_o \cup M_t - {}_{i,j}}$, which gives us a measure of the amount of unwanted change for C. Additionally, if the total number of changed pixels exceeds 500, the score for the main category is set to zero.

The final Filtering Augmentation Score (FAS) is composed from a multiplication of S_c and a binary score for the

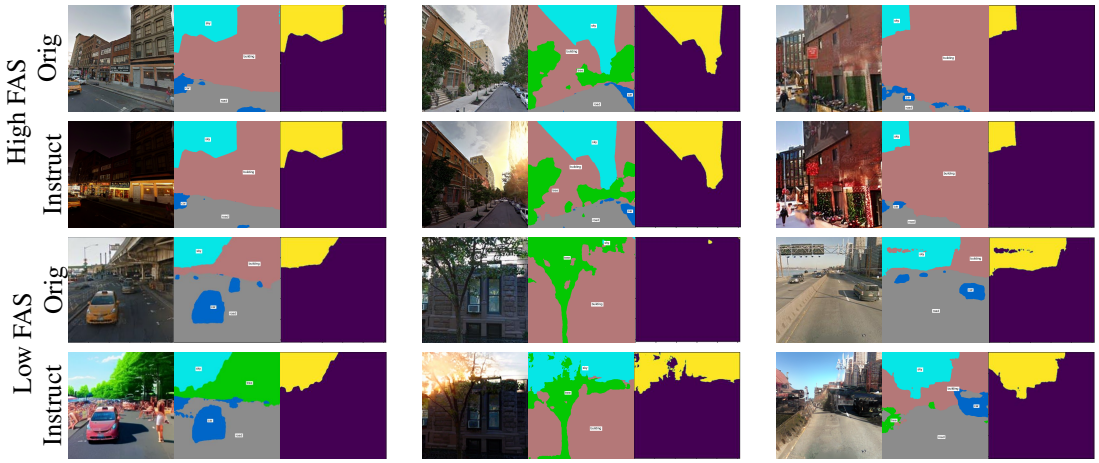

Figure 2. **Filtering InstructPix2Pix Image augmentations.** The ΔIm training set was filtered using our Filtering Augmentation Score (FAS), detailed in Section 2.5. Above, we present three examples with high (top rows) and low (bottom rows) FAS values. Each example shows two rows: the original image with its segmentation map and binary mask (M_0), followed by the InstructPix2Pix transformation with its corresponding segmentation and mask (M_t). As illustrated above, these binary masks provide meaningful cues for whether or not the model's output modifies the scene structure, which is undesirable for our problem setting.

secondary categories. We assign a binary score to the secondary categories to evaluate if they have undergone a significant change. If a second category's mask exceeds 10,000 pixels and more than 50% of the pixels have changed, we assign a score of 0; otherwise, we assign a score of 1. We empiricially set the FAS threshold to 0.92.

In Figure 2, we provide three examples of InstructPix2Pix transformations with high FAS and three with low FAS (*i.e.*, which are excluded from the final training set). For each example, the top row respectively shows the original image, the reduced label segmentation map, and M_o. The second row displays the image after InstructPix2Pix transformation, its reduced label segmentation map, and M_t. The left column on the top row shows a transformation to a night time scene. Note that SegFormer effectively identifies the building despite the darkness, and M_o is closely matches M_t. The middle column illustrates a transformation to sunrise. In the top example, SegFormer accurately identified the building lines despite that severe transformation. However, in the bottom example, the transformation was too extreme, turning the building into sky (which is accurately reflected in both M_o and M_t). In the right column, the top example demonstrates that the transformation into a Christmas scene altered the scene (and the original segmentation map). However, by reducing the labels, it becomes clear that the relevant semantic regions remained unchanged. In the bottom right example, InstructPix2Pix algorithm's transformation failed, and on the bottom left example the transformation to summer changed the structure of the scene. Both correctly identified by reduced label segmentation map. Note that although the change of the skyline (the change between M_o and M_t) in the bottom left example is not drastic, it is accurately assigned a low FAS score.

2.6. Baselines

We compare our method with six pre-existing methods for rotation extraction, including both classical (SIFT) and learning-based methods (LoFTR, 8PointVit, ExtremeRotation, CascadedAtt, Dust3R). The methods estimate the pose for each pair, and the evaluation metrics are calculated from the pose estimation, as described in the main paper. For SIFT and LoFTR, the pose estimation might fail due to an insufficient number of points that pass the geometric method (RANSAC). These pairs are excluded from the evaluation metric statistics, as is done in previous works. It is important to mention that this exclusion provides an advantage to these methods, as removing the invalid pairs likely reduces the number of "hard" pairs, thereby increasing their corresponding metric statistics. The percentage of invalid pairs is presented to capture this phenomenon.

SIFT The SIFT pipeline uses the OpenCV Python library, and the default random number generator is initialized with the seed value 12345. Images are resized so that their largest dimension is 256, and their intrinsic matrices are adjusted accordingly. Keypoints and descriptors for both grayscale images are detected using the SIFT detector. The 'Flann' KNN algorithm is used to match the key points, followed by filtering matches using Lowe's ratio test, ensuring that the nearest distance is smaller than the next nearest by a factor of 0.7. Image pairs with less than 6 matches are filtered. Since image pairs can have different intrinsic camera

	Method	StreetLearn			*s*ELP			*w*ELP		
		MGE↓	RRA_{15}↑	RRA_{30}↑	MGE↓	RRA_{15}↑	RRA_{30}↑	MGE↓	RRA_{15}↑	RRA_{30}↑
Large	SIFT [9]	0.10	100.0	100.0	2.14	91.5	94.1	4.26	70.8	79.1
	LoFTR [13]	0.11	100.0	100.0	1.84	98.6	99.6	2.47	90.6	96.6
Small	SIFT [9]	0.36	96.1	98.2	4.98	67.0	72.0	9.51	57.8	68.0
	LoFTR [13]	0.48	100.0	100.0	2.58	94.4	98.4	5.57	77.7	92.6
None	SIF* [9]	97.94	36.2	38.1	140.22	0.0	1.8	152.33	3.2	6.4
	LoFTR [13]	-	-	-	-	-	-	14.10	60.0	100.0

Table 2. **Homography-based estimation for feature-based techniques**. We evaluate performance over the *s*ELP and *w*ELP test sets, separately considering Large (top), Small (middle) and Non-overlapping (bottom) pairs. Note that median errors are computed only over successful image pairs, for which these algorithms output a pose estimate (failure over more than 50% of the test pairs is shown in gray).

parameters, the keypoints are adjusted so that their intrinsic camera parameters are represented by an identity matrix for essential matrix calculation. Essential matrix is then calculated using RANSAC confidence = 0.999, threshold = 0.01. Finally, the pose is recovered from the essential matrix and the valid matched key points. The success rates for large, small, none categories for *s*ELP test set: 91.4%, 58.6%, 5.7% and for *w*ELP test set: 65.6%, 43.2%, 7.7% .

LoFTR The LoFTR pipeline also utilizes the OpenCV Python library, and the default random number generator is initialized with the seed value 12345. Images are resized so that their largest dimension is 256, and their intrinsic matrices are adjusted accordingly. Key points and descriptors for both grayscale images are detected using the LOFTR pretrained model('outdoor_ds'). Image pairs with less than 20 matches are filtered. Since image pairs can have different intrinsic camera parameters, the keypoints are adjusted so that their intrinsic camera parameters are represented by an identity matrix for essential matrix calculation. Essential matrix is then calculated using RANSAC confidence = 0.999, threshold = 0.01. Finally, the pose is recovered from the essential matrix and the valid matched key points. The success rates for large, small, none categories for *s*ELP test set: 97.0%, 39.0%, 0.0% and for *w*ELP test set: 82.9%, 33.3%, 0.46% .

In Table 2 in the main body of the paper, we followed prior works to calculate the essential matrix for SIFT and LoFTR. We also conducted an experiment to evaluate these methods using homography, with the results presented in Table 2. We observe that homography-based estimation yields improvements over the StreetLearn dataset where over our real-world test sets, the transformation appears to cause distruptions, as evidenced by the increase in the MGE of SIFT for Large overlap from 2.1° to 2.5° in *w*ELP.

8PointViT The 8PointVit pipeline uses the 'streetlearn' pretrained model. Images are cropped around the center to a square dimension and resized to (256,256). The pose is then evaluated using the 8PointVit network.

ExtremeRotation The ExtremeRotation pipeline uses the 'streetlearn_cv_distribution' pretrained model. Images are cropped around the center to a square dimension and resized to (256,256). The relative Euler angles are then evaluated using the classification network.

CascadedAtt Since the model of CascadedAtt[4] was not released, we compare to it by directly implementing its encoder using weight-sharing Siamese residual U-nets, followed by cross-decoding with a weight-sharing transformer. These features are concatenated with Euler angle position embeddings and processed by our Rotation Estimation Transformer module, with the output rotation represented as relative Euler angles. We compare performance on the Streetlearn dataset evaluated in their work, finding that our re-implementation is mostly comparable, achieving only slightly lower MGE scores.

Dust3R The checkpoint utilized was the 512 DPT head, which yielded superior outcomes compared to the Dust3R checkpoints. The optimization technique that led to the pose estimation is PnP RANSAC. The pair is fed into the model in two instances, with their positions reversed in each case, resulting in two separate estimations of relative rotation. The chosen relative rotation is the one associated with the greatest confidence score, which is determined by the product of the average values from the pair's confidence maps.

3. Additional Results and Visualizations

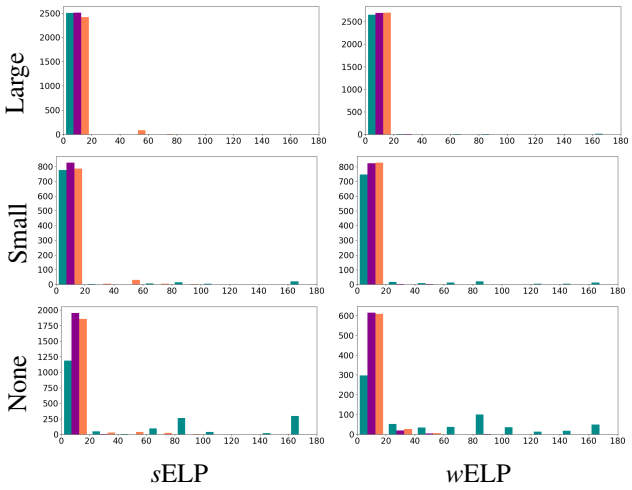

Figure 3. **Histograms over the yaw, pitch and roll relative rotation errors**. As illustrated above, yaw errors contribute significantly to the total error count. As detailed in [3], these yaw errors typically signify the uncertainties associated with non-overlapping pairs. Notably, the error peaks tend to occur at intervals that are multiples of 90 degrees.

Evaluation on Images Cropped from Panoramas. In Table 3, we conduct an evaluation over panoramic perspec-

Method	Large		Small		None	
	MGE ↓	RRA_{10} ↑	MGE ↓	RRA_{10} ↑	MGE↓	RRA_{10} ↑
DenseCorrVol [3]	1.09	99.4	1.41	98.3	**1.53**	**96.4**
CascadedAtt [4]	1.42	**100.0**	1.89	98.3	2.06	96.2
8PointViT [11]	**0.52**	**100.0**	**1.07**	**100.0**	101.51	6.0
Ours	1.06	**100.0**	1.09	**100.0**	1.98	**96.4**

Table 3. **Evaluation on Images Cropped from Panoramas**. We compare results over the Streetlearn panoramas test set, first reported in Cai *et al.* [3]. Note that all models (including ours) were trained on the same data; for this comparison, we report the performance of the model obtained after the *initialization* stage.

tive images, using the training and test set reported in prior work [3, 4]. Note that all models are trained and evaluated on StreetLearn [10] images; no data augmentations or additional data sources are used for this evaluation. As illustrated in the table, our model yields comparable performance over such constrained image pairs, achieving state-of-the-art relative rotation accuracy for non-overlapping pairs, matching the performance reported in prior work.

Top 5 Another measure used in this paper is *Top 5*, where output rotation matrix is selected from the best of 5 picks in the 360-D yaw angle output (and the top1 pick for pitch and roll). Figure 3 shows the geodesic error histograms for each relative rotation angle. The analysis reveals that yaw errors contribute significantly to the total error count. As detailed in [3], these yaw errors typically signify the uncertainties associated with non-overlapping pairs. Notably, the error peaks tend to occur at intervals that are multiples of 90 degrees.

In Figure 4, we demonstrate the top-5 predictions of our model for several non-overlapping pairs from the *w*ELP test set with high geodesic top-1 errors. To identify the *Top 5* predictions, we start with the 360-D yaw angle output, and apply a softmax function to it. Next, we smooth the distribution by applying a Gaussian kernel on it (using a standard deviation of 5). Finally, we locate the top 5 local maximas on the smoothed output.

We also examined the behavior of our model on pairs sampled from entirely different places. We conducted an experiment over 200 such image pairs, sampled from different scenes. Their histogram (Figure 5) reveals that the overall angle spans almost the entire range, with a noticeable tendency to larger angles (the overall angle of 64.5% of the pairs is above 90 degrees).

4. Additional Ablations

We report results over the *s*ELP test set for the ablations discussed in the main paper. Table 4 shows the effect of our progressive training scheme, while Table 5 demonstrates the impact of adding additional channels. As illustrated, these results are consistent with the results reported in the main paper over the *w*ELP test set. Next we provide additional

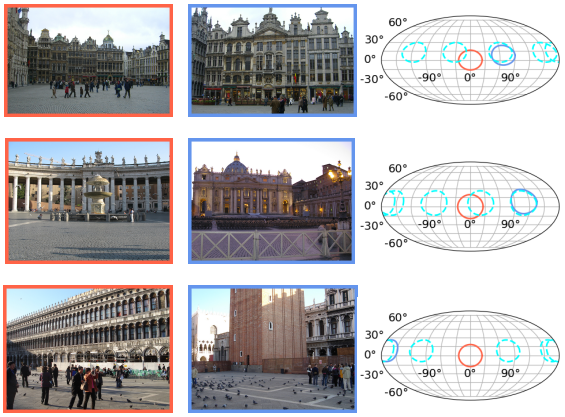

Figure 4. **Visualizing the top-5 predictions for non-overlapping pairs with high geodesic top-1 errors**. As illustrated by the examples above and also in our quantitative ablations, valuable insights can be found not only in the top predictions, but also among the subsequent selections. Images on the left serve as the reference points, and their coordinate system detemines the relative rotation, which defines the images on the right. The ellipsoids representing the ground truth are color-coded to match their respective images, with the estimated relative rotation illustrated by a cyan dashed line.

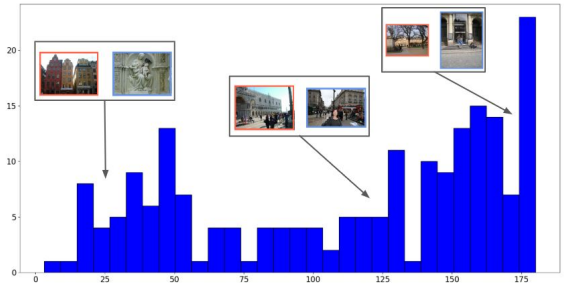

Figure 5. **Histogram of overall angle estimation of 200 randomly sampled image pairs from different scenes**.

ablations which further ablate various design considerations and our progressive training scheme.

4.1. The impact of our progressive training scheme

To further evaluate the efficacy of our progressive training approach, we compared it to training on all pairs simultaneously. We used a pre-trained model initially trained on panoramic images with a 90 degree field of view (FoV). All-at-once model (indicated by $\vee$ was then further trained on a combined dataset comprising ΔFoV, ΔIm, and *Extreme-LandmarkPairs* training sets, using three separate batches for each dataset. The results are presented in Table 4, and demonstrate that our progressive training approach is particularly beneficial in challenging scenarios, such as when there is no overlap with a single camera (*s*ELP test set), and

	Overlap	ΔFoV	ΔIm	*ELP*	Top 1 MGE↓	Top 1 RRA_{15}↑	Top 1 RRA_{30} ↑	Top 5 MGE↓	Top 5 RRA_{15}↑	Top 5 RRA_{30}↑
*s*ELP	Large	×	×	×	8.17	86.4	97.1	8.17	87.5	98.1
		∨	∨	∨	2.87	97.7	98.8	2.82	**99.2**	**99.8**
		✓	×	×	3.18	98.2	98.2	3.21	97.9	98.7
		✓	✓	×	3.10	**99.4**	**99.5**	4.04	98.5	98.9
		✓	✓	✓	**2.45**	96.7	96.8	**2.44**	96.8	97.0
	Small	×	×	×	26.44	6.3	60.2	22.32	17.4	69.5
		∨	∨	∨	5.72	87.7	**92.1**	5.56	**92.6**	**97.7**
		✓	×	×	5.14	79.8	82.6	5.24	85.5	92.1
		✓	✓	×	5.91	83.9	88.4	5.84	88.3	95.5
		✓	✓	✓	**4.35**	**88.3**	89.0	**4.15**	91.6	93.7
	None	×	×	×	76.63	17.9	32.9	17.54	42.0	78.9
		∨	∨	∨	23.92	41.5	52.2	11.51	64.7	87.7
		✓	×	×	28.22	42.4	49.7	10.52	71.0	90.8
		✓	✓	×	14.20	50.1	60.1	9.76	72.7	92.6
		✓	✓	✓	**13.62**	**52.7**	**59.7**	**9.25**	**75.4**	**93.6**
*w*ELP	Large	×	×	×	13.65	35.4	73.5	12.22	61.4	84.7
		∨	∨	∨	2.76	97.0	98.2	2.72	97.9	**99.6**
		✓	×	×	4.61	79.7	81.1	4.41	90.3	98.9
		✓	✓	×	4.46	90.4	92.4	4.43	94.3	99.1
		✓	✓	✓	**2.41**	**97.5**	**97.9**	**2.41**	**98.4**	99.4
	Small	×	×	×	55.28	3.7	29.1	29.83	15.0	50.3
		∨	∨	∨	6.46	81.9	90.1	5.99	86.1	**97.7**
		✓	×	×	12.91	56.2	68.2	10.97	66.0	85.4
		✓	✓	×	11.46	62.5	80.6	10.73	68.0	91.0
		✓	✓	✓	**4.47**	**87.2**	**91.6**	**4.24**	**91.1**	97.2
	None	×	×	×	74.94	12.8	25.3	25.11	26.1	58.8
		∨	∨	∨	65.74	24.1	37.7	16.73	45.1	72.6
		✓	×	×	61.62	25.0	38.4	16.82	44.2	75.0
		✓	✓	×	68.31	25.0	36.1	16.21	45.7	78.2
		✓	✓	✓	**26.97**	**36.1**	**50.7**	**12.85**	**57.1**	**85.8**

Table 4. Ablation study, evaluating the effect of our progressive training scheme over the two test sets (*s*ELP in the top rows, and *w*ELP in the bottom rows). All experiments start with the cropped panoramas used in Cai *et al.* [3]. We also assess the necessity of the progressive training scheme compared to training on all pairs simultaneously. Comparison denoted with ∨ is training with all of the datasets together. Best results are in bold.

for the *w*ELP test set. This method helps the model gradually adapt to the diverse challenges posed by real-world images. Another interesting finding from this ablation is the improved generalization capabilities of our model, trained only on 90 degrees FOV panorama perceptive images, in comparison to prior work [3] for in-the-wild image pairs. This result supports our decision to switch to a LoFTR encoder that was trained on Internet images, which could narrow the disparity between real-life images and panorama crops.

To assess the contribution of our components, we conducted an ablation study removing both InstructPix2Pix and SegFormer networks (Table 6), finding that results are slightly worse in this case. For example, for non-overlapping pairs the MGE is 46.08°. Importantly, even without these additional cues, our method still outperforms the baselines in the challenging no-overlapping scenario, demonstrating that while these networks do enhance our results, our improved performance is not solely dependent on these additional cues.

	Overlap	KP	SM	Top 1 MGE↓	Top 1 RRA_{15}↑	Top 1 RRA_{30} ↑	Top 5 MGE↓	Top 5 RRA_{15}↑	Top 5 RRA_{30}↑
*s*ELP	Large	×	×	**2.27**	97.9	98.4	**2.27**	98.0	98.6
		×	✓	2.31	96.8	97.1	2.28	98.0	99.0
		✓	×	2.40	**99.5**	**99.7**	2.40	**99.5**	**99.8**
		✓	✓	2.45	96.7	96.8	2.44	96.8	97.0
	Small	×	×	12.72	91.1	92.7	4.56	94.0	97.1
		×	✓	4.07	87.8	89.1	**4.00**	91.8	96.1
		✓	×	**4.04**	**92.9**	**93.8**	**4.00**	**96.3**	**98.8**
		✓	✓	4.35	88.3	89.0	4.15	91.6	93.7
	None	×	×	22.85	42.9	51.0	10.48	66.6	86.6
		×	✓	16.41	47.9	54.5	10.26	68.7	85.5
		✓	×	**12.92**	**55.0**	**63.2**	9.43	**76.8**	**95.3**
		✓	✓	13.62	52.7	59.7	**9.25**	75.4	93.6
*w*ELP	Large	×	×	**2.18**	97.4	98.1	**2.18**	97.4	98.1
		×	✓	2.30	97.0	97.4	2.30	**98.5**	99.4
		✓	×	2.44	**97.6**	**98.3**	2.31	98.4	**99.7**
		✓	✓	2.41	97.5	97.9	2.41	98.4	99.4
	Small	×	×	4.50	87.9	91.6	4.50	87.9	91.7
		×	✓	4.49	**88.1**	92.0	4.46	91.2	96.7
		✓	×	**4.41**	87.5	**92.2**	4.32	**91.9**	**97.6**
		✓	✓	4.47	87.2	91.6	**4.24**	91.1	97.2
	None	×	×	48.81	34.0	44.1	**12.56**	**57.5**	84.6
		×	✓	43.07	31.2	44.2	13.99	53.5	83.2
		✓	×	41.39	35.3	46.8	13.04	56.9	**86.2**
		✓	✓	**26.97**	**36.1**	**50.7**	12.85	57.1	85.8

Table 5. Ablation study, evaluating the effect of the auxiliary channels added as input to our network. We train models without the keypoints and matches (KP) and without the segmentation maps (SM), and compare with our full model that is provided with both.

Overlap	ΔIm	SM	Top 1 MGE↓	Top 1 RRA_{15}↑	Top 1 RRA_{30} ↑	Top 5 MGE↓	Top 5 RRA_{15}↑	Top 5 RRA_{30}↑
Large	×	×	2.43	97.3	98.1	2.34	98.0	99.5
	×	✓	**2.40**	**97.6**	98.1	2.39	**98.5**	99.4
	✓	×	2.44	**97.6**	**98.3**	**2.31**	98.4	**99.7**
	✓	✓	2.41	97.5	97.9	2.41	98.4	99.4
Small	×	×	4.64	86.7	90.6	4.44	90.3	97.2
	×	✓	4.50	87.2	92.0	4.34	91.3	**97.8**
	✓	×	**4.41**	**87.5**	**92.2**	4.32	**91.9**	97.6
	✓	✓	4.47	87.2	91.6	**4.24**	91.1	97.2
None	×	×	46.08	35.4	46.1	13.35	55.8	83.8
	×	✓	40.01	36.1	50.2	13.21	56.3	84.4
	✓	×	41.39	35.3	46.8	13.04	56.9	**86.2**
	✓	✓	**26.97**	**36.1**	**50.7**	**12.85**	**57.1**	85.8

Table 6. Ablation study, evaluating the effect of the priors of heavy neural networks - InstructPix2Pix and SegFormer.

To demonstrate that the improved performance cannot be simply obtained with the *ExtremeLandmarkPairs* training set, we report performance obtained without training on images cropped from panoramas in Table 7. We also compare performance to models trained on multiple stages of our progressive training scheme, followed by training on the *ExtremeLandmarkPairs* dataset as the final step. As illustrated in Table 7, relying solely on real image pairs from the *ELP*

	Overlap	[3]	ΔFoV	ΔIm	Top 1 MGE↓	Top 1 RRA_{15}↑	Top 1 RRA_{30}↑	Top 5 MGE↓	Top 5 RRA_{15}↑	Top 5 RRA_{30}↑
*s*ELP	Large	×	×	×	11.89	63.8	97.3	11.28	68.3	98.8
		✓	×	×	3.12	98.9	99.5	2.96	99.2	99.6
		✓	✓	×	**2.42**	96.3	96.4	**2.42**	96.4	96.7
		✓	✓	✓	2.45	**96.7**	**96.8**	2.44	**96.8**	**97.0**
	Small	×	×	×	37.39	14.4	36.9	18.61	38.9	77.5
		✓	×	×	6.34	83.0	89.6	5.84	88.3	**96.4**
		✓	✓	×	4.39	87.3	88.4	4.24	90.9	93.5
		✓	✓	✓	**4.35**	**88.3**	**89.0**	**4.15**	**91.6**	93.7
	None	×	×	×	107.20	5.5	13.5	23.31	29.5	63.3
		✓	×	×	24.26	37.6	53.8	12.29	61.3	84.8
		✓	✓	×	14.58	50.8	58.4	9.47	74.6	93.2
		✓	✓	✓	**13.62**	**52.7**	**59.7**	**9.25**	**75.4**	**93.6**
*w*ELP	Large	×	×	×	10.07	69.5	95.9	9.47	75.1	98.7
		✓	×	×	3.35	94.2	96.5	3.18	95.2	97.5
		✓	✓	×	**2.40**	**97.6**	**98.1**	**2.39**	**98.5**	**99.4**
		✓	✓	✓	2.41	97.5	97.9	2.41	98.4	**99.4**
	Small	×	×	×	30.26	21.5	49.3	14.39	52.2	83.6
		✓	×	×	9.48	67.8	83.7	8.34	72.7	90.8
		✓	✓	×	4.50	**87.2**	92.0	4.34	**91.3**	**97.8**
		✓	✓	✓	**4.47**	**87.2**	**91.6**	**4.24**	91.1	97.2
	None	×	×	×	75.44	4.8	18.8	23.58	26.0	63.1
		✓	×	×	48.77	24.1	39.3	16.43	46.7	79.9
		✓	✓	×	29.74	**36.1**	50.2	13.21	56.3	84.4
		✓	✓	✓	**26.97**	**36.1**	**50.7**	**12.85**	**57.1**	**85.8**

Table 7. We evaluate to what extent the progressive training scheme is needed, in contrast to simply training on real image pairs. In the first rows (three ×'s), we train models on *Extreme-LandmarkPairs only*. In the rows below, we first train on the panoramas, according to our progressive training scheme. As illustrated above, training on panoramas, along with the various data augmentations we propose, significantly improve performance, particularly for non-overlapping image pairs.

Method	Large MGE↓	Large RRA_{10}↑	Small MGE↓	Small RRA_{10}↑	None MGE↓	None RRA_{10}↑
Ours w/o LT	**0.98**	**100.0**	**1.06**	98.7	2.61	87.7
Ours	1.06	**100.0**	1.09	**100.0**	**1.98**	**96.4**

Table 8. **Network Architecture Ablation**. We evaluate the impact of the learnable tokens (LT) added to our model, comparing performance over the StreetLearn dataset. Best results are in bold.

train set yields significant performance degradation, particularly for extreme scenarios. Initialization with images cropped from panoramas improves performance. In particular, for large overlap on the *s*ELP test set, it achieves the best RRA percentages. However, as overlap between the images decreases, the errors remain relatively high. Our data augmentations allow for further improving performance, by creating diverse perspective images that better resemble the real samples.

	Method	*s*ELP MGE↓	*s*ELP RRA_{15}↑	*s*ELP RRA_{30}↑	*w*ELP MGE↓	*w*ELP RRA_{15}↑	*w*ELP RRA_{30}↑
Large	DenseCorrVol [3]	4.40	86.3	91.7	5.21	74.5	82.3
	Ours	**2.41**	**96.1**	**96.1**	**2.41**	**97.5**	**97.9**
Small	DenseCorrVol [3]	15.4	49.2	59.2	63.54	22.2	32.1
	Ours	**4.27**	**87.4**	**88.4**	**4.47**	**87.2**	**91.6**
None	DenseCorrVol [3]	103.97	17.9	29.4	95.46	11.2	19.0
	Ours	**14.16**	**51.4**	**58.4**	**26.97**	**36.1**	**50.7**

Table 9. **Finetune baseline with ΔIm and ΔFoV.** We evaluate performance over the *s*ELP and *w*ELP test sets, separately considering Large (top), Small (middle) and Non-overlapping (bottom) pairs.

4.2. Architectural Ablations

We conduct an ablation to evaluate the impact of using learnable tokens. As illustrated in Table 8, the performance without learnable tokens significantly deteriorates for non overlapping pairs. Specifically, RRA_{10} decreases from 96.4% to 87.7% and the median increases from 1.98 to 2.61.

We further investigated architectural differences through an additional ablation where we applied our progressive training scheme to the baseline models (Table 9). This experiment further shows that prior models are not directly applicable for real-world settings, as these baselines perform significantly worse across all metrics, particularly for Small and Non-overlapping cases. For instance, DenseCorrVol yields a MGE of 63.54° (Small) and 95.46° (None).

References

[1] Roberto Cipolla Alex Kendall, Matthew Grimes. Posenet: A convolutional network for real-time 6-dof camera relocalization. In *ICCV -International Conference on Computer Vision*, pages 2938–2946, 2015. 1

[2] Tim Brooks, Aleksander Holynski, and Alexei A. Efros. Instructpix2pix: Learning to follow image editing instructions. In *CVPR IEEE / CVF Computer Vision and Pattern*, pages 18392–18402, 2023. 3

[3] Ruojin Cai, Bharath Hariharan, Noah Snavely, and Hadar Averbuch-Elor. Extreme rotation estimation using dense correlation volumes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14566–14575, 2021. 2, 3, 5, 6, 7, 8

[4] Shay Dekel, Yosi Keller, and Martin Cadik. Estimating extreme 3d image rotations using cascaded attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2588–2598, 2024. 5, 6

[5] Xie Enze, Wang Wenhai, Yu Zhiding, Anandkumar Anima, M. Alvarez Jose, and Luo Ping. Segformer: Simple and efficient design for semantic segmentation with transformers. In *Conference on Neural Information Processing Systems (NEURIPS)*, 2021. 1, 2, 3

[6] Yicong Hong, Kai Zhang, Jiuxiang Gu, Sai Bi, Yang Zhou, Difan Liu, Feng Liu, Kalyan Sunkavalli, Trung Bui, and Hao Tan. Lrm: Large reconstruction model for single image to 3d. *arXiv preprint arXiv:2311.04400*, 2024. 2

[7] Zhengqi Li and Noah Snavely. Megadepth: Learning single-view depth prediction from internet photos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2041–2050, 2018. 1, 2

[8] Jin Linyi, Zhang Jianming, Wang Yannick, Hold-Geoffroyand Oliver, Sticha Kevin, Blackburn-Matzenand Matthew, and F. Fouhey David. Perspective fields for single image camera calibration. In *Computer Vision and Pattern Recognition Conference (CVPR)*, pages 17307–17316, 2023. 1

[9] David G Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004. 5

[10] Piotr Mirowski, Andras Banki-Horvath, Keith Anderson, Denis Teplyashin, Karl Moritz Hermann, Mateusz Malinowski, Matthew Koichi Grimes, Karen Simonyan, Koray Kavukcuoglu, Andrew Zisserman, et al. The streetlearn environment and dataset. *arXiv preprint arXiv:1903.01292*, 2019. 6

[11] Chris Rockwell, Justin Johnson, and David F Fouhey. The 8-point algorithm as an inductive bias for relative pose prediction by vits. In *2022 International Conference on 3D Vision (3DV)*, pages 1–11. IEEE, 2022. 6

[12] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 1

[13] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. Loftr: Detector-free local feature matching with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8922–8931, 2021. 5

[14] Joseph Tung, Gene Chou, Ruojin Cai, Guandao Yang, Kai Zhang, Gordon Wetzstein, Bharath Hariharan, and Noah Snavely. Megascenes: Scene-level view synthesis at scale. *arXiv preprint arXiv:2406.11819*, 2024. 1

[15] Peng Wang, Hao Tan, Sai Bi, Yinghao Xu, Fujun Luan, Kalyan Sunkavalli, Wenping Wang, Zexiang Xu, and Kai Zhang. Pf-lrm: Pose-free large reconstruction model for joint pose and shape prediction. *arXiv preprint arXiv:2311.12024*, 2023. 2

[16] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. *arXiv preprint arXiv:2312.14132*, 2023. 3

[17] Changchang Wul. Towards linear-time incremental structure from motion. In *International Conference on 3D Vision*, 2013. 1