

Learning Global Camera Poses from Noisy View-Graphs for Structure from Motion

Fadi Khatib[✉], Meirav Galun[✉], and Ronen Basri[✉]

Weizmann Institute of Science

Project webpage: vgpa-sfm.github.io

Abstract. Camera pose estimation is a key step in 3D reconstruction and view-synthesis pipelines. We present a deep, global Structure-from-Motion framework based on learned view-graph aggregation. Our method employs a permutation-equivariant, edge-conditioned graph neural network that takes noisy pairwise relative poses as input and outputs globally consistent camera extrinsics. The network is trained without ground-truth supervision, relying solely on a relative-pose consistency objective. This is followed by 3D point triangulation and robust bundle adjustment. Our approach is efficient, scalable to more than a thousand images, and robust to graph density. We evaluate our method on MegaDepth, 1DSfM, Strecha, and BlendedMVS. These experiments demonstrate that our method achieves superior rotation and translation accuracy compared to deep track-centric methods while registering more images across many scenes, and competitive results compared to state-of-the-art classical pipelines, while being much faster.

Keywords: Structure-from-Motion · Camera Pose Estimation

1 Introduction

Camera pose recovery is an essential part of 3D scene reconstruction and view synthesis applications. Many common Multiview Stereo (MVS) [35, 50] and view synthesis methods, including Neural Radiance Fields (NeRF) [26] and Gaussian Splatting (GS) [17] rely on accurate camera poses computed in preprocessing. View synthesis methods, in particular, have gained much popularity in recent years, as they can produce novel, realistically looking images and walkthroughs for complex scenes.

Multiview Structure-from-Motion (SfM) techniques provide reliable tools for camera pose recovery. Sequential pipelines, e.g., COLMAP [34], solve for one camera at a time, enriching the recovered set of camera poses and 3D points by processing image by image. These, generally highly accurate techniques, are relatively slow when applied to large collections of images, and their performance depends on the order in which the images are processed. Projective factorization techniques [40] simultaneously solve for all cameras and point tracks. These methods, however, attempt to factor large tensors that include all the track points.

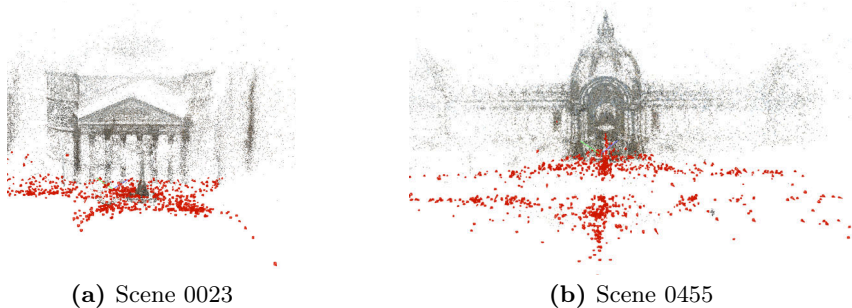


Fig. 1: 3D reconstructions and recovered camera parameters produced by VGPA on two large scenes ($N_c > 1000$ images). VGPA registers almost all images and scales to thousand-image collections, in contrast to existing image-based deep methods (e.g., VGGT, VGGSfM).

In the past decade, *global methods* emerged as an alternative to sequential and factorization methods. Global methods use a technique called *motion averaging*; given pairwise relative camera motion measurements, they seek to recover the location and orientation (and possibly also the intrinsic parameters) of cameras in a global coordinate system. Typically, this is done by solving separately for rotations and translations [28, 41], while some recent works developed techniques for directly averaging essential and fundamental matrices [15, 16]. Global methods can be more efficient than both sequential and factorization-based techniques, as they only solve for pose and therefore do not need to access and manipulate point tracks, except in the final bundle adjustment (BA) step.

In this paper, we reexamine the use of global SfM through the lens of *learned view-graph aggregation*. Specifically, we propose an efficient permutation-equivariant, edge-conditioned graph neural network (GNN) that takes as input noisy estimates of pairwise relative camera poses associated with the edges of a view graph, and outputs globally consistent camera extrinsics. The network is trained *without ground-truth supervision* using only a relative-pose consistency objective. Unlike existing deep-based approaches to SfM [8, 18, 27], our pose regression network does not use point tracks; it does not predict 3D points and does not rely on a reprojection loss. At test time, we use our network to predict global camera poses. Then, we improve our camera pose predictions by triangulating point tracks and applying robust BA. Finally, an optional and efficient *view reintegration* step is applied to recover cameras that were discarded in the process by the pipeline, increasing camera coverage.

Our approach is efficient and achieves high accuracy. It copes well with large-scale inputs, including scenes with more than a thousand images. In addition, our method is not restricted to exhaustive pairwise matching. We also evaluate a sparse retrieval-based view graph constructed from the top-30 MegaLoc [4] neighbors, and observe that VGPA often preserves strong registration coverage and accuracy despite the large reduction in edge density. We note that in the un-

calibrated setting, we optimize jointly for the intrinsics and extrinsics parameters during BA.

We perform an extensive experimental evaluation on challenging datasets, including MegaDepth and 1DSfM. These experiments demonstrate that our learned pose averaging achieves lower camera position and orientation errors than existing deep track-centric methods while registering more images on many scenes [8, 18, 27], and is competitive with strong classical pipelines. Similar results are obtained on smaller calibrated benchmarks for which ground truth measurements are available (Strecha and BlendedMVS) and on scenes containing challenging cyclic trajectories, where reprojection-centric methods such as [8, 18, 27] often struggle.

Below we summarize our contributions.

- We introduce **VGPA**, a permutation-equivariant GNN that performs *View-Graph Pose Averaging*, learning to aggregate noisy pairwise relative poses to recover globally consistent camera extrinsics.
- VGPA achieves accurate camera pose and structure recovery, matching the accuracy of state-of-the-art classical SfM pipelines while **being significantly faster**, and substantially outperforming recent feed-forward deep reconstruction methods on large-scale scenes.
- We train VGPA in a **self-supervised manner** by enforcing relative-pose consistency only, while 3D structure is recovered through triangulation followed by robust bundle adjustment.
- We show that VGPA can operate on sparse retrieval-based view graphs, obtaining competitive results when exhaustive pairwise matching is replaced by a top- k retrieval graph, despite large differences in edge density.

2 Related work

A popular classical method for Structure-from-Motion (SfM) uses an incremental algorithm in which images are processed one at a time, gradually extending the recovered set of camera poses and 3D structure. [1, 34, 37, 48]. While these methods achieve highly accurate reconstruction, they are inefficient when applied to large image collections, and their results depend on the order in which images are processed.

A second approach uses projective factorization to solve simultaneously for camera pose and 3D structure on all input images [12, 23, 40]. This method uses the observation that point track matrices are rank 4 when the points are scaled properly. Classical algorithms based on SVD factorization, however, are restricted to uncalibrated settings and do not handle missing data or outliers. Inspired by these techniques, several recent works train equivariant network architectures to jointly estimate camera poses and 3D structure from point tracks [8, 10, 18, 27]. These methods use either set-of-sets or graph transformer network architectures and are trained with either supervised or unsupervised data. An inlier/outlier classifier is incorporated for improved robustness [18].

Accurate pose recovery results were achieved with this method. However, it tends to over-prune valid inliers, leading to occasional registration failures and reduced image coverage.

Our method follows a third approach, commonly referred to as a *global approach*. Global methods handle all images simultaneously by applying manifold averaging to ensure the consistency of pairwise pose relations (rotations and translations) inferred from the essential matrices. Existing methods commonly solve first for camera orientations, and next for location and scales [25, 28, 30, 41]. [15, 16] introduced an averaging method for averaging essential and fundamental matrices, solving for all of these parameters in a single optimization. With the exception of [31], these methods require a separate step of 3D point triangulation. Theia [41] and the recent GLOMAP [31], in particular, were shown to yield accurate recovery.

Several recent works train networks to solve rotation averaging on the view graph. NeurORA [33] learns to denoise pairwise relative rotations and aggregates them to recover global orientations, while [21] applies message passing on pose graphs to iteratively update node rotations. These methods only address rotation averaging; they are trained on supervised data and tested in limited settings that do not include cross-dataset generalization. In contrast, our method is trained with unsupervised data and recovers the full camera extrinsics.

Recent learnable SfM methods such as VGGSfM [43], DUST3R [46], and MAST3R [20] are restricted to processing only a small number of input images, whereas Ace-Zero [6] and FlowMap [36] are tailored for video sequences under constant illumination. More recently, VGGT [42] introduced an end-to-end transformer that jointly predicts camera poses, dense 3D structure, and point tracks. Although promising, VGGT requires substantial supervised training and is currently restricted to images on the order of ~ 200 .¹ Fast3R [49] scales to larger collections but typically attains lower accuracy than VGGT at comparable settings. Continuous 3D perception methods such as CUT3R [45] maintain a persistent scene state and incrementally predict camera parameters and pointmaps from an image stream, while TTT3R [9] extends this framework with test-time training to improve long-sequence reconstruction without modifying the model parameters. However, while such methods scale to long sequences, they perform poorly on large unordered image collections.

In this paper, we introduce a learned view-graph pose averaging module implemented with a permutation-equivariant graph neural network. Trained without ground-truth supervision, our method achieves competitive accuracies at lower runtime than strong global SfM baselines and surpasses prior deep factorization approaches in both accuracy and camera coverage.

¹ In May 2026, after our experiments were completed, the VGGT authors reported an implementation fix that removes redundant intermediate tensors kept in memory, allowing the model to process roughly $2\text{--}3\times$ more frames under the same GPU memory budget.

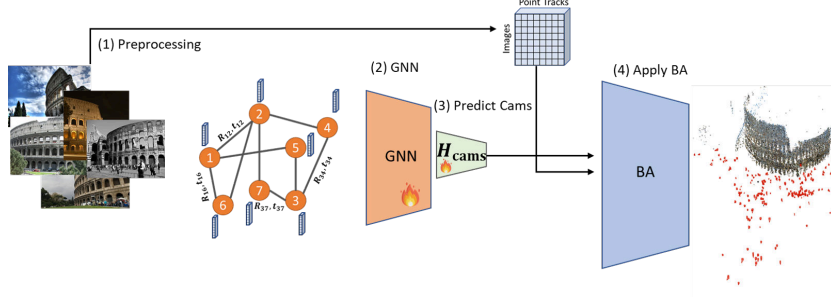


Fig. 2: Method overview. (1) *Preprocessing*: estimate pairwise relative poses from essential matrices and build the view graph; extract point tracks. (2) *GNN*: a permutation-equivariant, edge-conditioned GNN aggregates the view graph to produce camera embeddings. (3) *Predict cams*: a small head H_{cams} regresses global extrinsics $(R_i, \mathbf{t}_i)$ from the embeddings. (4) *Triangulation + BA*: using the predicted cameras and the point tracks, we triangulate 3D points and run robust bundle adjustment.

3 Method

Given a collection of m images of a stationary scene, we assume, as in standard SfM pipelines, that in preprocessing we extract (1) essential matrices and (2) a collection of point tracks, which will form the input to our pipeline. Our objective is to recover the camera matrices for all the given images and a triangulated 3D location for each track. Below, we describe each step in our method.

3.1 Preprocessing

Denote our input images by $I_1, \dots, I_m$. Following standard SfM pipelines, we begin by detecting and matching features across the images using SIFT [24]. We next apply RANSAC [5] and obtain a partial collection of pairwise essential matrices $\{E_{ij}\}_{i,j \in [m]}$, denoted by $\mathcal{E}$. Each essential matrix encodes the relative rotation R_{ij} and translation $\mathbf{t}_{ij}$ between camera P_i and P_j . We extract the rotation and translation by decomposing the essential matrix, while enforcing positive depth. Note that $\mathbf{t}_{ij}$ is determined at this point only up to scale. These pairwise rotation and translation measurements serve as input to our pose averaging module.

A second outcome of the procedure above comprises pairs of matched feature points across images. We next use heuristics (as in, e.g., [34]) to join such pairs to form longer tracks. Each track is a set $T_k = \{\mathbf{x}_{i_1,k}, \mathbf{x}_{i_2,k}, \dots\}$ with $i_1, i_2, \dots \in [m]$, and we assume that T_k contains the projected locations of a single 3D scene point, denoted $\mathbf{X}_k$, onto $I_{i_1}, I_{i_2}, \dots$. These tracks are generally contaminated by small displacement errors (noisy measurements) and outliers.

We will use this collection of point tracks at a later stage to triangulate the 3D structure using the predicted absolute camera poses.

3.2 Network architecture

Our network applies *pose averaging* to the view graph. As is shown in Fig. 2, it comprises two modules: (i) a permutation-equivariant, edge-conditioned GNN that aggregates pairwise relative poses into camera embeddings; and (ii) a regression head that predicts global camera parameters from these embeddings. **Pose-averaging GNN.** We build a viewing graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ whose nodes index the m images and whose edges carry relative-pose measurements. For each edge $(i, j) \in \mathcal{E}$ we define

$$\mathbf{e}_{ij} = \phi_e(\log R_{ij}^{\text{RANSAC}}, \mathbf{t}_{ij}^{\text{RANSAC}}),$$

where $\log R_{ij}^{\text{RANSAC}} \in \mathbb{R}^3$ is the $\text{SO}(3)$ log-map of the relative rotation recovered from RANSAC and $\mathbf{t}_{ij}^{\text{RANSAC}} \in \mathbb{S}^2$ is the unit translation direction recovered from the essential matrix, and ϕ_e is a learned MLP. Since the available geometric information resides on the edges of the view graph, node embeddings are initialized with a shared learnable token $\mathbf{h}_i^{(0)} = \mathbf{h}_0$.

We apply edge-conditioned message passing with degree-normalized mean aggregation:

$$\begin{aligned} \tilde{\mathbf{m}}_i^{(\ell)} &= \sum_{j \in \mathcal{N}(i)} \phi_m(\mathbf{h}_i^{(\ell)}, \mathbf{h}_j^{(\ell)}, \mathbf{e}_{ij}), \\ \mathbf{m}_i^{(\ell)} &= \frac{1}{|\mathcal{N}(i)|} \tilde{\mathbf{m}}_i^{(\ell)}, \\ \mathbf{h}_i^{(\ell+1)} &= \text{LN}(\mathbf{h}_i^{(\ell)} + \text{Drop}(\psi(\text{LN}(\mathbf{h}_i^{(\ell)}), \mathbf{m}_i^{(\ell)}))), \end{aligned}$$

for $\ell = 0, \dots, L-1$, where ϕ_m and ψ are MLPs, LN denotes Layer Normalization [3], and Drop denotes Dropout [38]. The network is equivariant to node relabelings (permutations) of $\mathcal{G}$. In ablation experiments we also evaluated replacing the message passing operator with a graph attention mechanism (GATv2), but observed no performance improvements while incurring higher computational cost. The final node embeddings are $z_i = \mathbf{h}_i^{(L)}$.

Pose regression head. The pose regression head obtains as input the per-camera embeddings $\mathbf{z}_i$ produced by the pose-averaging GNN. A 3-layer MLP head H_{cams} maps these embeddings to camera parameters,

$$(\mathbf{t}_i, \mathbf{q}_i) = H_{\text{cams}}(\mathbf{z}_i), \quad \tilde{\mathbf{q}}_i \leftarrow \mathbf{q}_i / \|\mathbf{q}_i\|,$$

where $\mathbf{t}_i \in \mathbb{R}^3$ and $\tilde{\mathbf{q}}_i \in \mathbb{H}$ is a unit quaternion.

3.3 Output and loss

Our network predicts the m internally calibrated cameras $P_1, \dots, P_m$. Each camera is parameterized as $P_i = [R_i \mid \mathbf{t}_i]$ with $R_i \in \text{SO}(3)$ and $\mathbf{t}_i \in \mathbb{R}^3$; the camera center is $-R_i^\top \mathbf{t}_i$.

Training is unsupervised and seeks cameras $P_1, \dots, P_m$ that best agree with the pairwise relative-pose estimates. We therefore minimize a relative-pose consistency objective. Specifically, we use

$$\mathcal{L}_{\text{RelPose}} = \frac{1}{|\mathcal{E}|} \sum_{(i,j) \in \mathcal{E}} d_R(\hat{R}_{ij}, R_{ij}^{\text{RANSAC}}) + \frac{1}{|\mathcal{E}|} \sum_{(i,j) \in \mathcal{E}} d_t(\hat{\mathbf{t}}_{ij}, \mathbf{t}_{ij}^{\text{RANSAC}}), \quad (1)$$

where $\hat{R}_{ij}$ and $\hat{\mathbf{t}}_{ij}$ denote the relative rotation and translation estimated from the output cameras P_i and P_j using

$$\hat{R}_{ij} = R_j R_i^\top, \quad \hat{\mathbf{t}}_{ij} = \mathbf{t}_j - R_j R_i^\top \mathbf{t}_i, \quad (2)$$

R_{ij}^{RANSAC} and $\mathbf{t}_{ij}^{\text{RANSAC}}$ are the corresponding rotation and translation obtained with RANSAC in preprocessing, $d_R(R_1, R_2) = \arccos\left(\frac{\text{trace}(R_1^\top R_2) - 1}{2}\right)$ is the geodesic rotation error, and $d_t(\mathbf{a}, \mathbf{b}) = \arccos\langle \mathbf{a}, \mathbf{b} \rangle$ measures directional disagreement.

Training protocol. We iterate over all training scenes in each epoch. Models are selected by early stopping on a held-out validation set; we report the checkpoint with the lowest validation error. Additional implementation details and architectural specifications are provided in the Supplementary Material.

Inference. On an *unseen* scene, the model predicts all camera poses in a single forward pass. We then fine-tune on the target scene with the unsupervised objective (no ground-truth labels) for $T_{\text{ft}} = 200$ steps. Next, we triangulate using DLT [13] to recover 3D point positions from the estimated camera poses and point tracks, and finally perform a robust bundle adjustment initialized with the camera poses predicted by the network and the triangulated points.

4 Experiments

4.1 Datasets

We train our network on scenes from the MegaDepth dataset [22] and then test it on a diverse range of real-world scenes that include novel scenes from the MegaDepth dataset as well as cross-dataset generalization tests on the 1DSfM dataset [47], Strecha [39], and BlendedMVS [51]. We refer the reader to the supplementary material for hyperparameters and further technical details.

MegaDepth [22]. The MegaDepth dataset includes 196 different outdoor landmark scenes curated from the internet. We followed the train/test split as in [18], including subsampling of scenes with more than 1000 images. In Table 1, above the middle rule are scenes with fewer than 1000 images, while the scenes below the rule are subsampled.

1DSfM [47]. 1DSfM is a collection of diverse urban scenes reconstructed from community photo collections. We use this dataset to test our method (trained on the MegaDepth dataset) in cross-dataset generalization experiments, demonstrating large-scale reconstructions in realistic settings.

Strecha [39]. The Strecha dataset consists of small outdoor scenes (≤ 30 images) and includes ground-truth data acquired with a LIDAR system.

BlendedMVS [51]. The BlendedMVS dataset includes synthetic scenes with textured meshes rendered and blended to produce color images and depth maps, providing ground truth camera poses.

Ground truth camera poses. Many challenging datasets, including MegaDepth and 1DSFM, lack ground-truth measurements; therefore, as is common in the field, we use camera poses computed with COLMAP [34], a state-of-the-art incremental Structure from Motion (SfM) method, to generate “ground truth” camera poses. COLMAP is widely used for this purpose (see [8, 11, 14, 18, 29, 47, 52]) due to its accurate and robust performance. To evaluate our method with real ground truth, we additionally show results on the smaller datasets Strecha [39] and BlendedMVS [51].

4.2 Baselines

With the exception of VGGT [42], the settings and results for all baselines below were taken from [18].

RESfM [18]. RESfM is a robust deep equivariant SfM model that operates on a point-track tensor using a sets-of-sets permutation-equivariant architecture. It augments prior equivariant factorization by adding a multiview inlier/outlier classifier integrated into the same equivariant backbone and concludes with a robust bundle-adjustment stage.

Theia [41]. A global SfM pipeline that applies rotation averaging, followed by translation averaging, and finally 3D point triangulation.

GLOMAP [31]. A global SfM pipeline that first applies rotation averaging, followed by an integrated step of translation averaging and point triangulation.

VGGSfM [44] is a differentiable, trainable SfM pipeline.

MASt3R [20]. An SfM pipeline that utilizes a global alignment procedure to merge pairwise pointmap predictions.

VGGT [42]. VGGT is a feed-forward, end-to-end multi-view transformer network that jointly predicts cameras, depth, point maps, and tracks for up to about 200 views. It uses alternating inter-frame/global attention and is additionally refined with BA.

We also evaluate several recent feed-forward 3D reconstruction models designed to scale to large image collections, including **FAST3R** [49], a transformer-based model that predicts per-pixel 3D locations in a shared reference frame in a single forward pass, and **CUT3R** and **TTT3R** [9, 45], continuous 3D perception models that incrementally reconstruct scene geometry and camera parameters from an image stream while maintaining a persistent scene representation.

4.3 Metrics and evaluation

To evaluate our results, we first align the predicted scenes to the ground truth by applying a per-scene 3D similarity transformation. We then compare our camera orientation predictions with the ground truth ones using angular differences in

Table 1: MegaDepth experiment. For each scene, we show the number of input images (denoted N_c) and the fraction of outliers. For each model, we show the number of images used for reconstruction (denoted N_r) and mean values of the rotation error (in degrees) and translation error. Above the middle rule are Group 1 scenes with < 1000 images; below are Group 2 scenes with > 1000 images, subsampled to 300 for testing. Winning results are marked in **bold and underlined**.

Scene	N_c	Outliers%	Ours			RESfM			Theia			GLOMAP		
			N_r	Rot	Trans	N_r	Rot	Trans	N_r	Rot	Trans	N_r	Rot	Trans
0238	522	44.6%	486	1.06	0.285	283	2.61	0.325	506	1.21	0.334	497	0.74	0.349
0060	528	41.6%	518	0.10	0.026	503	0.29	0.029	525	0.85	0.124	520	0.11	0.048
0197	870	40.7%	718	0.58	0.108	667	4.22	0.333	855	1.16	0.227	813	0.43	0.130
0094	763	40.1%	659	0.81	0.116	537	3.77	0.750	742	0.75	0.160	711	0.88	3.907
0265	571	38.8%	372	5.30	1.433	346	1.25	0.389	554	5.83	2.216	554	7.46	2.839
0083	635	31.3%	622	0.32	0.062	596	0.64	0.058	632	0.37	0.372	614	0.08	0.016
0076	558	30.5%	547	0.09	0.037	524	0.37	0.094	549	0.78	0.120	540	0.17	0.042
0185	368	30.0%	364	0.12	0.021	350	0.06	0.010	365	0.41	0.094	365	0.16	0.051
0048	512	24.2%	501	0.10	0.011	474	4.69	0.178	507	0.41	0.105	505	0.15	0.224
0024	356	23.0%	328	3.59	1.122	309	2.03	0.398	355	0.56	0.219	338	0.15	0.104
0223	214	17.0%	211	3.45	0.285	204	3.76	0.510	212	3.34	0.519	213	1.75	0.275
5016	28	16.9%	28	0.08	0.015	28	0.12	0.016	28	0.10	0.061	28	0.08	0.046
0046	440	14.6%	438	0.47	0.073	399	0.95	0.043	434	0.25	0.112	440	0.03	0.007
0099	299	47.4%	157	0.56	0.229	190	3.53	0.709	297	3.28	0.664	255	0.15	0.085
1001	285	43.9%	268	3.87	2.585	251	1.70	0.661	275	7.89	3.990	270	4.56	3.817
0231	296	42.2%	260	0.31	0.029	246	0.84	0.065	286	1.37	0.322	278	0.73	0.134
0411	299	29.9%	268	0.23	0.036	273	0.13	0.020	293	0.39	0.196	268	0.19	0.148
0377	295	27.5%	232	0.12	0.016	210	0.29	0.018	269	1.13	0.205	266	0.65	0.237
0102	299	25.8%	296	0.18	0.023	284	0.28	0.059	294	2.31	0.698	293	0.15	0.101
0147	298	24.6%	289	1.36	0.118	207	4.62	0.325	284	6.36	0.934	289	6.75	3.542
0148	287	24.6%	244	1.15	0.135	197	0.60	0.035	275	13.98	1.558	282	22.73	2.646
0446	298	22.1%	296	0.34	0.023	288	0.71	0.046	289	1.23	0.391	295	0.20	0.071
0022	297	21.2%	280	0.11	0.015	274	0.29	0.039	296	0.58	0.160	280	0.22	0.087
0327	298	21.0%	280	0.48	0.032	271	0.26	0.090	288	1.27	0.360	288	15.54	2.035
0015	284	20.6%	241	0.61	0.083	215	1.04	0.167	244	2.21	0.389	272	0.28	0.095
0455	298	19.8%	292	0.40	0.071	293	0.68	0.105	294	0.77	0.159	297	0.35	0.064
0496	297	19.2%	280	0.69	0.041	281	0.35	0.055	285	1.40	0.550	291	0.44	0.303
1589	299	17.4%	294	0.13	0.074	290	0.14	0.019	288	0.82	0.193	298	0.07	0.041
0012	299	16.3%	295	0.77	0.086	287	0.40	0.027	129	1.04	0.318	295	0.51	0.121
0104	284	16.2%	237	4.04	0.330	193	0.29	0.029	265	17.05	1.530	280	19.69	0.834
0019	299	15.4%	288	0.23	0.008	250	0.06	0.008	271	0.81	0.250	296	0.09	0.025
0063	293	14.5%	266	0.14	0.024	262	0.46	0.048	268	0.92	0.605	287	0.32	0.100
0130	285	14.4%	202	0.21	0.019	192	0.20	0.023	187	1.20	0.349	279	2.00	0.909
0080	284	12.9%	141	0.10	0.017	139	0.59	0.096	278	2.62	0.868	282	1.92	0.236
0240	298	11.9%	288	0.64	0.088	275	3.13	0.265	285	1.31	0.470	290	0.39	0.135
0007	290	11.7%	284	1.51	0.079	172	0.91	0.041	277	1.24	0.174	289	0.19	0.035

degrees. We measure differences between our predicted and ground truth camera locations using the l_2 distance. For a fair comparison, both our method and all baseline methods operating on point tracks were evaluated using the same set of point tracks. For all methods, we apply a final post-processing step of robust bundle adjustment. Additional tables reporting AUC metrics are provided in the supplementary.

4.4 Results

Our results on the MegaDepth and 1DSfM test sets and comparisons to baselines are shown in Tables 1 and 2, respectively. For each scene, we also report the number of input images (N_c), the fraction of outlier track points, and compare our VGPA method against the baselines in terms of number of registered images, mean rotation error (in degrees), translation error, and runtime.

Across both benchmarks, VGPA outperforms the deep factorization baseline RESfM on most scenes, achieving lower rotation and translation errors. Com-

pared to classical pipelines, VGPA is competitive with Theia and GLOMAP, and often surpasses them on both metrics. In terms of coverage, VGPA registers a larger fraction of images than RESfM, though typically fewer than GLOMAP.

Table 2: 1DSfM experiment. For each scene, we show the number of input images (denoted N_c) and the fraction of outliers. For each model, we show the number of images used for reconstruction (denoted N_r) and mean values of the rotation error (in degrees) and translation error. Winning results are marked in **bold and underlined**.

Scene	N_c	Outliers%	Ours			RESfM			Theia			GLOMAP		
			N_r	Rot	Trans	N_r	Rot	Trans	N_r	Rot	Trans	N_r	Rot	Trans
Alamo	573	32.6%	523	1.35	0.322	484	3.66	0.515	549	4.42	1.433	557	2.45	1.520
Ellis Island	227	25.1%	215	0.28	0.081	214	0.82	0.122	213	5.01	1.527	219	0.58	0.155
Madrid Metropolis	333	39.4%	298	1.31	0.145	244	8.42	0.827	319	2.61	0.903	320	1.22	0.242
Montreal Notre Dame	448	31.7%	442	0.93	0.418	346	2.82	0.352	420	4.47	1.285	444	0.60	0.211
NYC Library	330	33.6%	295	0.34	0.095	224	3.96	0.429	313	4.06	1.141	323	0.58	0.189
Notre Dame	549	35.6%	527	0.52	0.105	517	1.20	0.231	531	3.70	0.828	543	2.73	0.389
Piazza del Popolo	336	33.1%	318	5.37	0.670	249	2.20	0.186	324	3.31	1.053	331	0.80	0.188
Tower of London	467	27.0%	457	0.89	0.057	94	0.67	0.026	448	6.61	1.189	466	0.81	0.138
Vienna Cathedral	824	31.4%	763	7.90	0.590	479	1.52	0.112	767	12.25	1.663	822	2.00	2.414
Yorkminster	432	29.0%	402	0.43	0.044	331	14.54	1.468	387	8.35	1.916	418	0.95	0.316

For completeness, we also report the performance of several recent deep learning-based pose estimation methods that aim to scale to larger image collections than those supported by approaches such as VGGT. The per-scene results are shown in Table 3. While these methods are computationally efficient, their pose accuracy on this benchmark remains noticeably lower than that of SfM-based pipelines.

Table 3: Deep-based methods on the 1DSfM dataset. For each scene we list the number of input images (N_c). For each deep model (TTT3R, CUT3R, FAST3R) we report the mean rotation error (degrees) and mean translation error. Best results are **bold and underlined**.

Scene	N_c	TTT3R		CUT3R		FAST3R	
		Rot	Trans	Rot	Trans	Rot	Trans
Alamo	573	16.08	3.650	22.32	4.266	40.37	3.990
Ellis Island	227	8.85	2.333	12.53	3.171	11.74	3.201
Madrid Metropolis	333	14.83	2.258	18.81	3.189	67.37	3.574
Montreal Notre Dame	448	13.25	1.230	16.12	2.134	10.79	2.052
NYC Library	330	7.67	1.656	10.26	1.912	11.90	2.408
Notre Dame	549	11.88	1.430	15.33	1.694	16.44	2.241
Piazza del Popolo	336	23.13	2.063	22.63	2.092	32.48	2.568
Tower of London	467	29.69	3.647	29.85	3.607	59.53	3.658
Vienna Cathedral	824	43.76	2.978	41.58	2.802	29.49	2.561
Yorkminster	432	16.45	2.223	23.00	2.992	20.26	2.463

Following [18], we evaluate VGPA on the smaller Strecha and BlendedMVS benchmarks, which provide ground-truth camera poses. As shown in Table 4, VGPA is consistently more accurate than image-based deep baselines (VGGsFm,

MASt3R, and VGGT), which typically do not scale to the larger datasets considered, and it performs on par with classical pipelines (including Theia, COLMAP, and GLOMAP).

Table 4: Strecha & BlendedMVS datasets. For each scene we list the number of input images (N_c) and outlier fraction. For each method we report the number of registered images (N_r), mean rotation error (deg), translation error, and runtime (s). Best is **bold**, second best is underlined.

Scene	N_c Out. %		Ours				VGGT				MASt3R				VGGSM				Theia				COLMAP				GLOMAP			
			N_r	Rot	Trans	Time	N_r	Rot	Trans	Time	N_r	Rot	Trans	Time	N_r	Rot	Trans	Time	N_r	Rot	Trans	Time	N_r	Rot	Trans	Time	N_r	Rot	Trans	Time
Strecha																														
entry-P10	10	4.8	10	0.004	0.0005	7	10	0.079	0.033	16.5	10	0.442	0.055	19	10	0.165	0.056	10.3	10	0.024	0.008	0.9	10	<u>0.023</u>	<u>0.007</u>	36.0	10	0.187	0.026	12.5
fountain-P11	11	1.4	11	0.009	0.0003	8	11	0.034	0.019	12.2	11	0.160	0.026	22	11	0.172	0.016	15.4	11	<u>0.027</u>	<u>0.002</u>	1.5	11	<u>0.027</u>	<u>0.003</u>	37.0	11	0.194	0.022	38.6
Herz-Jesus-P8	8	1.8	8	0.009	0.0010	6	8	0.032	0.011	12.7	8	0.363	0.037	16	8	0.206	0.042	8.7	8	<u>0.025</u>	0.005	0.6	8	0.026	<u>0.004</u>	22.0	8	0.091	0.015	<u>5.0</u>
Herz-Jesus-P25	25	2.8	25	0.010	0.0003	12	25	0.048	0.007	31.9	25	0.869	0.057	81	25	0.158	0.046	19.6	25	<u>0.026</u>	<u>0.006</u>	2.4	25	0.028	<u>0.006</u>	60.0	25	0.138	0.013	76.6
BlendedMVS																														
scene0	75	2.0	74	0.149	0.0204	73	75	0.041	0.017	108	75	0.501	0.191	516	75	0.045	0.0106	<u>61</u>	75	0.009	0.0017	49	75	0.006	0.0005	106	75	<u>0.007</u>	<u>0.0016</u>	198
scene1	51	1.4	51	0.342	0.0345	28	51	0.101	0.050	41	51	0.919	0.173	1017	51	0.098	0.0112	32	51	0.029	<u>0.0029</u>	18	51	0.007	0.0003	67	51	<u>0.024</u>	0.0102	117
scene2	33	2.2	33	<u>0.008</u>	<u>0.0004</u>	<u>17</u>	33	0.230	0.022	52	33	1.972	0.130	117	33	0.227	0.0180	30	33	0.045	0.0098	15	33	0.003	0.0002	55	33	0.025	0.0060	87
scene3	66	8.8	66	<u>0.006</u>	<u>0.0003</u>	54	66	0.353	0.014	276	66	0.927	0.045	815	66	0.372	0.0174	<u>52</u>	66	0.019	0.0018	21	66	0.004	0.0002	128	66	0.008	0.0017	392

Robustness to retrieval-based graph construction. We train VGPA using relative poses obtained from **exhaustive** pairwise matching. At test time, we additionally evaluate a sparse retrieval-based view graph, where each image is connected to its top-30 neighbors retrieved by MegaLoc [4]. As shown in Table 5, VGPA often preserves strong registration coverage and accuracy despite the large reduction in edge density.

Uncalibrated image collections. Table 6 compares two settings: (i) using ground-truth intrinsics and (ii) starting from an approximate calibration (f_x, f_y proportional to image size, principal point at the image center) and optimizing intrinsics jointly with the extrinsics during bundle adjustment. More details on this protocol are provided in the supplementary material. While self-calibration incurs a small accuracy drop relative to ground-truth intrinsics, VGPA remains competitive and maintains high performance.

View Re-integration (optional post-processing). The results reported in the main paper are obtained *without* this step. As an optional post-processing stage, we attempt to re-register views that may be discarded during the BA stage using a lightweight add-back procedure. Unregistered views are ranked by connectivity (e.g., number of 2D–3D matches) with the current point cloud. For each candidate, we estimate its pose from the available 2D–3D correspondences and refine it with a short local BA applied to its neighboring views. The process repeats until no further views can be added. Additional results comparing *Ours* and *Ours + post-processing* in terms of the number of registered images (N_r), mean rotation error (deg), and mean translation error are reported in the Supplementary Material. These results show that the add-back step increases the number of registered cameras with minimal runtime overhead (about 0.6 second per added view, without additional optimization).

Qualitative results. Table Fig. 1 shows 3D reconstructions and camera parameters obtained by VGPA for two scenes with more than 1,000 images; in both scenes, we register almost all images. These results demonstrate that our method

Table 5: Robustness to retrieval-based graph construction. We evaluate VGPA on the full 1DSfM benchmark using either exhaustive pairwise matching or a sparse retrieval-based view graph constructed with MegaLoc top-30 retrieval. For each scene, we report the number of input images (N_c), the number of registered images (N_r), mean rotation error (deg), and mean translation error. Best results are marked in **bold and underlined**.

Scene	N_c	Ours					
		Exhaustive Matching			MegaLoc@30		
		N_r	Rot	Trans	N_r	Rot	Trans
Alamo	573	523	1.35	0.322	548	1.18	0.163
Ellis Island	227	215	0.28	0.081	216	0.21	0.090
Madrid Metropolis	333	298	1.31	0.145	323	1.48	3.397
Montreal Notre Dame	448	442	0.93	0.418	338	0.11	0.015
NYC Library	330	295	0.34	0.095	319	0.92	0.537
Notre Dame	549	527	0.52	0.105	535	0.42	0.086
Piazza del Popolo	336	318	5.37	0.670	329	0.57	0.064
Tower of London	467	457	0.89	0.057	447	2.14	0.408
Vienna Cathedral	824	763	7.90	0.590	771	9.32	0.462
Yorkminster	432	402	0.43	0.044	405	6.32	0.469

Table 6: Impact of Camera Intrinsics (Known vs. Estimated). For each scene, we report the number of input images (N_c) and the outlier fraction. We compare our method with known intrinsics vs. without intrinsics and report N_r , mean rotation error (deg), and mean translation error. Best results are in **bold**.

Scene	N_c	Out.%	Ours (w/ intrinsics)			Ours (w/o intrinsics)		
			N_r	Rot	Trans	N_r	Rot	Trans
<i>BlendedMVS scenes (shared intrinsics)</i>								
scene0	75	2.0	74	0.149	0.0204	74	0.122	0.0159
scene1	51	1.4	51	0.342	0.0345	51	0.337	0.0401
scene2	33	2.2	33	0.008	0.0004	33	0.025	0.0050
scene3	66	8.8	66	0.006	0.0003	66	0.010	0.0014
<i>MegaDepth scenes (not shared intrinsics)</i>								
0012	299	16.3	295	0.77	0.086	292	0.98	0.253
0024	356	23.0	328	3.59	1.122	308	1.79	0.508
0048	512	24.2	501	0.10	0.011	485	0.38	0.456
0083	635	31.3	622	0.32	0.062	596	0.64	0.058

produces superior reconstructions and effectively handles outliers compared to the baselines. Moreover, VGPA is not limited by the number of images, unlike image-based deep methods such as VGGT and VGGSfM. Additional qualitative results on diverse scenes are shown in Fig. 3

Runtime. Table7 reports runtimes using the identical point tracks produced by our preprocessing. For a fair comparison, the reported runtime for VGPA includes all stages after preprocessing: network inference, test-time fine-tuning, triangulation, and the final robust BA refinement. VGPA is substantially faster than COLMAP, GLOMAP, and Theia, and achieves higher throughput. Importantly, these gains come without sacrificing reconstruction quality: VGPA achieves accuracy and coverage comparable to classical pipelines, demonstrating that learned view-graph pose averaging is efficient at scale.

Memory scalability. Our GNN operates directly on the edge set of the view graph, storing one feature vector per edge rather than materialising a full $N \times N$

Table 7: Runtime. Given the same point tracks, we compare the runtime of our proposed method (VGPA) to RESfM and classical methods, including COLMAP, Theia, and GLOMAP. We report N_r/t to normalize runtime with respect to the number of registered images.

Scene	N_c	Outliers%	Ours			RESfM			COLMAP			Theia			GLOMAP		
			Total (Mins)	N_r	$N_r/t \uparrow$	Total (Mins)	N_r	$N_r/t \uparrow$	Total (Mins)	N_r	$N_r/t \uparrow$	Total (Mins)	N_r	$N_r/t \uparrow$	Total (Mins)	N_r	$N_r/t \uparrow$
Alamo	573	32.6	3.9	523	134.1	17.2	484	28.2	83.7	568	6.8	13.4	553	41.4	40.0	557	13.9
Ellis Island	227	25.1	0.9	215	239.3	2.8	214	75.9	14.9	223	15.0	1.1	213	193.6	7.7	219	28.6
Madrid Metropolis	333	39.4	1.3	298	233.4	5.8	244	42.1	25.1	323	12.9	—	—	—	7.1	320	45.2
Montreal Notre Dame	448	31.7	2.1	442	206.4	6.1	346	56.7	35.9	447	12.5	3.7	422	114.6	13.5	444	32.9
Notre Dame	549	35.6	3.3	527	160.2	22.2	517	23.3	72.6	546	7.5	11.6	534	46.0	21.1	543	25.8
NYC Library	330	33.6	1.9	295	159.0	4.0	224	55.7	26.6	330	12.4	1.5	314	204.2	7.3	323	44.5
Piazza del Popolo	336	33.1	1.0	318	304.0	2.7	249	92.6	9.6	334	34.9	3.0	325	108.8	5.9	331	56.0
Tower of London	467	27.0	3.6	457	127.9	5.9	94	15.9	65.0	467	7.2	3.1	448	142.5	23.5	466	19.8
Vienna Cathedral	824	31.4	7.2	763	106.2	23.9	479	20.0	98.9	824	8.3	11.2	772	68.8	41.6	822	19.8
Yorkminster	432	29.0	2.9	402	137.2	7.7	331	42.9	31.4	419	13.3	2.9	390	135.3	14.8	418	28.2
Mean	—	—	2.8	424	180.8	9.8	318	45.3	46.4	448	13.1	5.7	441	117.2	18.2	444	31.5

attention matrix. Memory therefore scales as $\mathcal{O}(|\mathcal{E}| \cdot d)$, where $|\mathcal{E}|$ is the number of edges and d is the feature dimension. For a sparse k -nearest-neighbour graph this is $\mathcal{O}(Nk)$, i.e. linear in the number of cameras, making the method practical for large scenes. Even under a complete graph ($|\mathcal{E}| = \mathcal{O}(N^2)$), the architecture remains tractable for the scene sizes encountered in practice, since no dense pairwise matrix is ever allocated.

Ablations. Ablations confirm that each core component of our method is important. Reducing the number of GNN layers degrades performance, indicating that deeper message passing is beneficial for aggregating global view-graph information. We also evaluate replacing our message-passing operator with a graph attention mechanism (GATv2), but observe no improvement. A randomly initialized model performs poorly, while fine-tuning from random initialization improves performance but remains worse than the full method. This demonstrates the importance of pretraining, which learns representations that generalize across scenes. Finally, combining pretraining with fine-tuning yields the best results across rotation, translation, and pose AUC. All results are reported *before* the final BA refinement; see Table 8.

Table 8: Ablation study reporting rotation, translation, and pose AUC@30 on the validation set, *before* final BA refinement. Higher is better.

	Rotation AUC@30 ($\uparrow$)	Translation AUC@30 ($\uparrow$)	Pose AUC@30 ($\uparrow$)
2 GNN layers	0.755	0.367	0.346
Graph attention (GATv2)	0.788	0.385	0.364
Pretrained model (without fine-tuning)	0.789	0.375	0.358
Random init	0.316	0.024	0.012
Fine-tuning only (random init)	0.728	0.377	0.352
Full model (pretrain + fine-tune)	0.849	0.402	0.389



Fig. 3: Example reconstructions from the proposed VGPA on various datasets.

5 Conclusion

We present VGPA, an unsupervised deep *pose-averaging* network for multiview SfM. The design includes a permutation-equivariant pose-averaging module that enforces consistency of pairwise rotations and translation directions through message passing over the view graph. Additional 3D point triangulation and robust BA refinement ensure high accuracy and recover the 3D structure. Across challenging benchmarks (including MegaDepth, 1DSfM), VGPA outperforms deep methods and remains competitive with strong classical pipelines while maintaining high camera coverage. It is also *fast*: on the same point tracks, VGPA is substantially faster than COLMAP and GLOMAP, and modestly faster than Theia, while scaling to large image collections. Finally, a lightweight view re-integration step can optionally recover a portion of the few remaining discarded views with negligible overhead.

Acknowledgement. Research was partially supported by the MBZUAI-WIS Joint Program for Artificial Intelligence Research, by Minerva, by the Israeli Council for Higher Education (CHE) via the Weizmann Data Science Research Center, and by research grants from the Estates of Tully and Michele Plesser and the Anita James Rosen and Harry Schutzman Foundations.

References

1. Agarwal, S., Furukawa, Y., Snavely, N., Simon, I., Curless, B., Seitz, S.M., Szeliski, R.: Building rome in a day. *Communications of the ACM* **54**(10), 105–112 (2011)
2. Agarwal, S., Mierle, K., Others: Ceres solver. <http://ceres-solver.org>
3. Ba, J.L., Kiros, J.R., Hinton, G.E.: Layer normalization. *arXiv preprint arXiv:1607.06450* (2016)
4. Berton, G., Masone, C.: Megaloc: One retrieval to place them all. In: *Proceedings of the Computer Vision and Pattern Recognition Conference*. pp. 2861–2867 (2025)
5. Bolles, R.C., Fischler, M.A.: A ransac-based approach to model fitting and its application to finding cylinders in range data. In: *IJCAI*. vol. 1981, pp. 637–643 (1981)
6. Brachmann, E., Wynn, J., Chen, S., Cavallari, T., Monszpart, Á., Turmukhambetov, D., Prisacariu, V.A.: Scene coordinate reconstruction: Posing of image collections via incremental learning of a relocalizer. *arXiv preprint arXiv:2404.14351* (2024)
7. Brody, S., Alon, U., Yahav, E.: How attentive are graph attention networks? *arXiv preprint arXiv:2105.14491* (2021)
8. Brynte, L., Iglesias, J.P., Olsson, C., Kahl, F.: Learning structure-from-motion with graph attention networks. *arXiv preprint arXiv:2308.15984* (2023)
9. Chen, X., Chen, Y., Xiu, Y., Geiger, A., Chen, A.: Ttt3r: 3d reconstruction as test-time training. *arXiv preprint arXiv:2509.26645* (2025)
10. Chen, Z., Li, J., Li, Q., Yang, B., Dong, Z.: Deepaat: Deep automated aerial triangulation for fast uav-based mapping (2024)
11. Cui, Z., Tan, P.: Global structure-from-motion by similarity averaging. In: *Proceedings of the IEEE International Conference on Computer Vision*. pp. 864–872 (2015)

12. Dai, Y., li, H., He, M.: Element-wise factorization for n-view projective reconstruction. pp. 396–409 (09 2010). https://doi.org/10.1007/978-3-642-15561-1_29
13. Hartley, R., Zisserman, A.: Multiple view geometry in computer vision. Cambridge university press (2003)
14. Jiang, N., Cui, Z., Tan, P.: A global linear method for camera pose registration. In: Proceedings of the IEEE international conference on computer vision. pp. 481–488 (2013)
15. Kasten, Y., Geifman, A., Galun, M., Basri, R.: Algebraic characterization of essential matrices and their averaging in multiview settings. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5895–5903 (2019)
16. Kasten, Y., Geifman, A., Galun, M., Basri, R.: Gpsfm: Global projective sfm using algebraic constraints on multi-view fundamental matrices. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3264–3272 (2019)
17. Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G.: 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.* **42**(4), 139–1 (2023)
18. Khatib, F., Kasten, Y., Moran, D., Galun, M., Basri, R.: Resfm: Robust deep equivariant structure from motion. In: The Thirteenth International Conference on Learning Representations (2025)
19. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014)
20. Leroy, V., Cabon, Y., Revaud, J.: Grounding image matching in 3d with mast3r. *arXiv preprint arXiv:2406.09756* (2024)
21. Li, X., Ling, H.: Pogo-net: Pose graph optimization with graph neural networks. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 5895–5905 (2021)
22. Li, Z., Snavely, N.: Megadepth: Learning single-view depth prediction from internet photos. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2041–2050 (2018)
23. Lin, Y., Yang, L., Lin, Z., Lin, T., Zha, H.: Factorization for projective and metric reconstruction via truncated nuclear norm. In: 2017 International Joint Conference on Neural Networks (IJCNN). pp. 470–477 (2017). <https://doi.org/10.1109/IJCNN.2017.7965891>
24. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. *International journal of computer vision* **60**(2), 91–110 (2004)
25. Martinec, D., Pajdla, T.: Robust rotation and translation estimation in multiview reconstruction. In: 2007 IEEE Conference on Computer Vision and Pattern Recognition. pp. 1–8. IEEE (2007)
26. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM* **65**(1), 99–106 (2021)
27. Moran, D., Koslowsky, H., Kasten, Y., Maron, H., Galun, M., Basri, R.: Deep permutation equivariant structure from motion. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5976–5986 (2021)
28. Moulon, P., Monasse, P., Perrot, R., Marlet, R.: Openmvg: Open multiple view geometry. In: International Workshop on Reproducible Research in Pattern Recognition. pp. 60–74. Springer (2016)
29. Ozyesil, O., Singer, A.: Robust camera location estimation by convex programming. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2674–2683 (2015)

30. Özyeşil, O., Voroninski, V., Basri, R., Singer, A.: A survey of structure from motion*. *Acta Numerica* **26**, 305–364 (2017)
31. Pan, L., Baráth, D., Pollefeys, M., Schönberger, J.L.: Global structure-from-motion revisited. In: *European Conference on Computer Vision (ECCV)* (2024)
32. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al.: Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems* **32** (2019)
33. Purkait, P., Chin, T.J., Reid, I.: Neurora: Neural robust rotation averaging. In: *European conference on computer vision*. pp. 137–154. Springer (2020)
34. Schönberger, J.L., Frahm, J.M.: Structure-from-motion revisited. In: *Conference on Computer Vision and Pattern Recognition (CVPR)* (2016)
35. Seitz, S.M., Curless, B., Diebel, J., Scharstein, D., Szeliski, R.: A comparison and evaluation of multi-view stereo reconstruction algorithms. In: *2006 IEEE computer society conference on computer vision and pattern recognition (CVPR'06)*. vol. 1, pp. 519–528. IEEE (2006)
36. Smith, C., Charatan, D., Tewari, A., Sitzmann, V.: Flowmap: High-quality camera poses, intrinsics, and depth via gradient descent. *arXiv preprint arXiv:2404.15259* (2024)
37. Snavely, N., Seitz, S.M., Szeliski, R.: Photo tourism: exploring photo collections in 3d. In: *ACM siggraph 2006 papers*, pp. 835–846 (2006)
38. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research* **15**(1), 1929–1958 (2014)
39. Strecha, C., Von Hansen, W., Van Gool, L., Fua, P., Thoennessen, U.: On benchmarking camera calibration and multi-view stereo for high resolution imagery. In: *2008 IEEE conference on computer vision and pattern recognition*. pp. 1–8. Ieee (2008)
40. Sturm, P., Triggs, B.: A factorization-based algorithm for multi-image projective structure and motion. In: *European conference on computer vision*. pp. 709–720. Springer (1996)
41. Sweeney, C., Hollerer, T., Turk, M.: Theia: A fast and scalable structure-from-motion library. In: *Proceedings of the 23rd ACM international conference on Multimedia*. pp. 693–696 (2015)
42. Wang, J., Chen, M., Karaev, N., Vedaldi, A., Rupprecht, C., Novotny, D.: Vggt: Visual geometry grounded transformer. In: *Proceedings of the Computer Vision and Pattern Recognition Conference*. pp. 5294–5306 (2025)
43. Wang, J., Karaev, N., Rupprecht, C., Novotny, D.: Visual geometry grounded deep structure from motion. *arXiv preprint arXiv:2312.04563* (2023)
44. Wang, J., Karaev, N., Rupprecht, C., Novotny, D.: Vggsfm: Visual geometry grounded deep structure from motion. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 21686–21697 (2024)
45. Wang, Q., Zhang, Y., Holynski, A., Efros, A.A., Kanazawa, A.: Continuous 3d perception model with persistent state. In: *Proceedings of the Computer Vision and Pattern Recognition Conference*. pp. 10510–10522 (2025)
46. Wang, S., Leroy, V., Cabon, Y., Chidlovskii, B., Revaud, J.: Dust3r: Geometric 3d vision made easy. *arXiv preprint arXiv:2312.14132* (2023)
47. Wilson, K., Snavely, N.: Robust global translations with 1dsfm. In: *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part III* 13. pp. 61–75. Springer (2014)

48. Wu, C.: Towards linear-time incremental structure from motion. In: 2013 International Conference on 3D Vision-3DV 2013. pp. 127–134. IEEE (2013)
49. Yang, J., Sax, A., Liang, K.J., Henaff, M., Tang, H., Cao, A., Chai, J., Meier, F., Feiszli, M.: Fast3r: Towards 3d reconstruction of 1000+ images in one forward pass. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 21924–21935 (2025)
50. Yao, Y., Luo, Z., Li, S., Fang, T., Quan, L.: Mvsnet: Depth inference for unstructured multi-view stereo. In: Proceedings of the European conference on computer vision (ECCV). pp. 767–783 (2018)
51. Yao, Y., Luo, Z., Li, S., Zhang, J., Ren, Y., Zhou, L., Fang, T., Quan, L.: Blended-mvs: A large-scale dataset for generalized multi-view stereo networks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 1790–1799 (2020)
52. Zhang, J.Y., Lin, A., Kumar, M., Yang, T.H., Ramanan, D., Tulsiani, S.: Cameras as rays: Pose estimation via ray diffusion. In: International Conference on Learning Representations (ICLR) (2024)

A Implementation details

Code and data. Our code and preprocessed data will be made publicly available.

Framework. We train and evaluate on NVIDIA A100 GPUs (80 GB). The implementation uses PyTorch [32] and the Adam optimizer [19] with gradient normalization.

Training. Each epoch iterates over all training scenes. A held-out validation set is used for early stopping. Validation and test evaluations use the complete view graph. Training on MegaDepth takes approximately 7 hours on a single A100. We fix the random seed to 20.

Architecture details. The encoder uses 3 edge-conditioned message-passing layers, with both node and edge features represented using 256-dimensional embeddings, and ReLU activations throughout. The camera head H_{cams} is a 3-layer MLP with hidden dimension 256.

Hyperparameter search. We sweep over (1) learning rate $\{10^{-2}, 10^{-3}, 10^{-4}\}$, (2) network width $\{128, 256, 512\}$ for the encoder and heads, and (3) number of layers $\{2, 3, 4, 5\}$.

Bundle adjustment. We use Ceres Solver [2] with a Huber loss (scale 0.1) for robustness, following [18]. In each BA round, we cap the number of iterations at 300 or stop earlier on convergence.

Uncalibrated reconstruction. For experiments where camera intrinsics are not provided, we initialize the calibration using a centered principal point and a focal length proportional to the image size, following common SfM practice. These initial intrinsics are used during preprocessing to estimate pairwise relative poses from the matched image pairs. The resulting relative poses are then used as input to VGPA, which predicts the camera extrinsics. In the final reconstruction stage, we jointly refine the camera extrinsics, 3D points, and intrinsic parameters using bundle adjustment. Thus, the approximate calibration is used only to initialize the pipeline, while the final intrinsics are optimized together with the recovered scene geometry. In the released code, we will also provide calibration averaging as an optional preprocessing step, which may improve robustness in uncalibrated settings.

B More details on the ablation

Attention-based architecture. For the attention-based ablation, we replace the original message-passing GNN with a GATv2-based [7] network while keeping the overall prediction pipeline unchanged. The input to the network is a graph whose nodes represent cameras and whose pairwise relative-pose measurements are represented by edges. Each relative measurement, consisting of a relative rotation R_{ij} and a normalized translation direction t_{ij} , is first mapped to a learned edge embedding using the same encoder as in the main model.

In the variant used in our experiments, we introduce *measurement nodes* in addition to camera nodes. Concretely, for each relative-pose measurement between cameras i and j , we create a measurement node initialized from the corresponding edge embedding. The resulting graph is therefore bipartite: each measurement node is connected to its two endpoint camera nodes. The bipartite edge attributes consist of the relative-pose embedding.

Message passing is performed by alternating two attention-based updates within each layer: a camera-to-measurement update and a measurement-to-camera update. Both updates use GATv2 convolutions, allowing the contributions of neighboring nodes

to be weighted adaptively through learned attention coefficients conditioned on node and edge features. Each attention block is followed by residual connections, layer normalization, dropout, and a feed-forward MLP, following a pre-normalized transformer-style design. After several such layers, the final camera-node embeddings are passed to the same regression head as in the base model to predict the camera poses.

C Constructing point tracks

We follow the preprocessing of [18] to construct point tracks; see their Appendix for full details.

Runtime protocol for learning-based baselines. We report runtimes using a consistent protocol across methods: runtime is measured after the point-track construction stage and therefore does not include the time required to construct point tracks. This is the same convention used for VGPA. For MAST3R, the reported runtime also excludes the pairwise matching stage, which is substantially more expensive than the sparse matching used in our preprocessing. VGGT uses the VGGSfM tracker to construct tracks; in our experiments, this tracking stage has runtime comparable to our track-construction pipeline, with our preprocessing being slightly faster. Therefore, the runtime comparison with VGGT, MAST3R, and VGGSfM follows a fair and consistent protocol, and does not overstate the efficiency of VGPA relative to these learning-based baselines.

D Additional results

Here we present the view reintegration results (Table 9). Tables 10 and 11 report the AUC (area under the recall curve) scores, computed using the maximum of the relative rotation and translation errors between every image pair, across different thresholds (in degrees), for both the MegaDepth and 1DSfM experiments.

In Tables 12 and 13 we additionally report a registration-aware AUC (RA-AUC), which accounts for both pose accuracy and camera registration coverage. Unlike the standard AUC computed only over registered cameras, RA-AUC is computed over all $\binom{N_c}{2}$ image pairs. For any pair containing at least one unregistered image, we assign an error of 180° , so that the pair contributes zero recall at all evaluated thresholds. Equivalently, if the original AUC is computed only over registered image pairs, the registration-aware score is

$$\text{RA-AUC} = \text{AUC} \cdot \frac{\binom{N_r}{2}}{\binom{N_c}{2}}.$$

This metric penalizes incomplete reconstructions and therefore better reflects the trade-off between accuracy and registration completeness.

Note on RESfM AUC results. To compute the AUC and registration-aware AUC metrics, we re-ran RESfM using the same evaluation protocol as the other methods. Since the final bundle-adjustment stage is not fully deterministic, the resulting RESfM numbers may differ slightly from the original numbers reported in the RESfM paper and used in our main paper tables. These differences are minor and do not affect the conclusions.

Table 9: MegaDepth: effect of view reintegration. We report the number of input images (N_c), outlier fraction, registered images (N_r), and mean rotation and translation errors for our method (*Ours*) and with the add-back postprocessing step (*Ours + Post-Processing*).

Scene	N_c	Outliers%	Ours			Ours + postprocessing		
			N_r	Rot	Trans	N_r	Rot	Trans
0238	522	44.6%	486	1.06	0.285	515	1.08	0.288
0060	528	41.6%	518	0.10	0.026	526	0.10	0.027
0197	870	40.7%	718	0.58	0.108	844	0.89	0.151
0094	763	40.1%	659	0.81	0.116	717	1.82	0.200
0265	571	38.8%	372	5.30	1.433	531	6.42	1.900
0083	635	31.3%	622	0.32	0.062	632	0.32	0.062
0076	558	30.5%	547	0.09	0.037	552	0.09	0.036
0185	368	30.0%	364	0.12	0.021	367	0.12	0.021
0048	512	24.2%	501	0.10	0.011	507	0.10	0.011
0024	356	23.0%	328	3.59	1.122	349	3.81	1.142
0223	214	17.0%	211	3.45	0.285	214	3.55	0.291
5016	28	16.9%	28	0.08	0.015	28	0.08	0.015
0046	440	14.6%	438	0.47	0.073	440	0.46	0.073
0099	299	47.4%	157	0.56	0.229	215	0.93	0.291
1001	285	43.9%	268	3.87	2.585	277	3.78	2.595
0231	296	42.2%	260	0.31	0.029	279	0.52	0.063
0411	299	29.9%	268	0.23	0.036	292	0.26	0.044
0377	295	27.5%	232	0.12	0.016	263	0.18	0.021
0102	299	25.8%	296	0.18	0.023	298	0.18	0.023
0147	298	24.6%	289	1.36	0.118	295	1.84	0.140
0148	287	24.6%	244	1.15	0.135	262	2.58	0.270
0446	298	22.1%	296	0.34	0.023	298	0.35	0.024
0022	297	21.2%	280	0.11	0.015	288	0.11	0.016
0327	298	21.0%	280	0.48	0.032	293	1.45	0.085
0015	284	20.6%	241	0.61	0.083	260	0.89	0.153
0455	298	19.8%	292	0.40	0.071	298	0.41	0.071
0496	297	19.2%	280	0.69	0.041	292	0.70	0.047
1589	299	17.4%	294	0.13	0.074	298	0.14	0.075
0012	299	16.3%	295	0.77	0.086	298	1.18	0.137
0104	284	16.2%	237	4.04	0.330	256	9.01	0.707
0019	299	15.4%	288	0.23	0.008	296	0.32	0.012
0063	293	14.5%	266	0.14	0.024	278	0.32	0.035
0130	285	14.4%	202	0.21	0.019	212	0.87	0.099
0080	284	12.9%	141	0.10	0.017	159	1.93	0.231
0240	298	11.9%	288	0.64	0.088	296	0.62	0.087
0007	290	11.7%	284	1.51	0.079	289	1.49	0.077
Mean	379	26.1%	327	0.95	0.215	348	1.36	0.264

Table 10: MegaDepth experiment (AUC). For each scene, we list the number of input images (N_c) and the fraction of outliers. For each model, we report the number of registered cameras (N_r) and the AUC values at different error thresholds (in degrees). Winning results are marked in **bold and underlined**.

Scene	N_c	Outliers%	Ours				RESFM				Theia				GLOMAP			
			N_r	@1	@5	@30	N_r	@1	@5	@30	N_r	@1	@5	@30	N_r	@1	@5	@30
0238	522	44.6%	486	0.427	0.648	0.887	354	0.547	0.811	0.916	506	0.063	0.495	0.863	497	0.298	0.653	0.884
0060	528	41.6%	518	0.811	0.935	0.982	491	0.762	0.906	0.969	525	0.300	0.702	0.912	520	0.676	0.893	0.969
0197	870	40.7%	718	0.394	0.771	0.944	706	0.643	0.865	0.954	855	0.086	0.505	0.872	813	0.555	0.817	0.940
0094	763	40.1%	659	0.704	0.879	0.949	539	0.365	0.533	0.749	742	0.325	0.708	0.904	711	0.468	0.772	0.921
0265	571	38.8%	372	0.000	0.004	0.181	457	0.129	0.641	0.911	554	0.000	0.005	0.267	554	0.000	0.001	0.165
0083	635	31.3%	622	0.852	0.945	0.981	572	0.855	0.959	0.989	632	0.504	0.817	0.948	614	0.765	0.935	0.987
0076	558	30.5%	547	0.765	0.925	0.980	512	0.732	0.921	0.986	549	0.133	0.599	0.897	540	0.510	0.830	0.963
0185	368	30.0%	364	0.547	0.954	0.985	345	0.812	0.956	0.992	365	0.285	0.742	0.937	365	0.641	0.889	0.970
0048	512	24.2%	501	0.838	0.955	0.991	485	0.800	0.941	0.976	507	0.397	0.784	0.949	505	0.698	0.907	0.975
0024	356	23.0%	328	0.513	0.749	0.827	308	0.522	0.829	0.948	355	0.153	0.555	0.873	338	0.362	0.719	0.931
0223	214	17.0%	211	0.599	0.827	0.916	202	0.483	0.756	0.864	212	0.014	0.342	0.783	213	0.330	0.703	0.908
5016	28	16.9%	28	0.790	0.928	0.984	28	0.355	0.499	0.741	28	0.413	0.793	0.952	28	0.508	0.835	0.960
0046	440	14.6%	438	0.928	0.971	0.985	388	0.607	0.687	0.737	434	0.530	0.861	0.965	440	0.895	0.977	0.996
0099	299	47.4%	157	0.239	0.637	0.869	222	0.048	0.319	0.636	297	0.011	0.172	0.619	255	0.526	0.769	0.909
1001	285	43.9%	268	0.018	0.314	0.697	256	0.005	0.241	0.779	275	0.000	0.001	0.051	270	0.000	0.003	0.133
0231	296	42.2%	260	0.599	0.867	0.968	247	0.542	0.855	0.954	286	0.063	0.467	0.842	278	0.417	0.762	0.921
0411	299	29.9%	268	0.692	0.910	0.977	273	0.693	0.917	0.984	293	0.188	0.600	0.902	268	0.380	0.727	0.928
0377	295	27.5%	232	0.793	0.943	0.986	197	0.769	0.938	0.989	269	0.198	0.596	0.882	266	0.567	0.824	0.941
0102	299	25.8%	296	0.763	0.922	0.979	280	0.679	0.906	0.977	294	0.169	0.474	0.785	293	0.547	0.805	0.946
0147	298	24.6%	289	0.635	0.778	0.924	241	0.416	0.552	0.694	284	0.055	0.468	0.771	289	0.000	0.000	0.000
0148	287	24.6%	244	0.504	0.698	0.926	256	0.416	0.546	0.672	275	0.050	0.280	0.499	282	0.296	0.472	0.591
0446	298	22.1%	296	0.677	0.896	0.972	288	0.627	0.891	0.973	289	0.053	0.460	0.840	295	0.465	0.799	0.947
0022	297	21.2%	280	0.706	0.925	0.987	273	0.676	0.916	0.985	296	0.194	0.631	0.900	280	0.473	0.797	0.949
0327	298	21.0%	280	0.767	0.922	0.974	270	0.751	0.931	0.983	288	0.040	0.538	0.884	288	0.474	0.683	0.776
0015	284	20.6%	241	0.767	0.896	0.951	223	0.547	0.718	0.808	244	0.134	0.478	0.784	272	0.572	0.839	0.950
0455	298	19.8%	292	0.735	0.892	0.961	293	0.712	0.916	0.979	294	0.209	0.644	0.905	297	0.585	0.862	0.966
0496	297	19.2%	280	0.721	0.909	0.966	281	0.615	0.876	0.963	285	0.080	0.498	0.849	291	0.441	0.786	0.942
1589	299	17.4%	294	0.654	0.898	0.971	290	0.744	0.930	0.987	288	0.157	0.480	0.797	298	0.583	0.812	0.951
0012	299	16.3%	295	0.786	0.907	0.957	291	0.682	0.825	0.862	129	0.087	0.499	0.840	295	0.645	0.887	0.971
0104	284	16.2%	237	0.590	0.698	0.846	195	0.727	0.932	0.988	265	0.127	0.430	0.642	280	0.445	0.634	0.717
0019	299	15.4%	288	0.855	0.950	0.987	247	0.832	0.962	0.993	271	0.243	0.649	0.888	296	0.740	0.924	0.986
0063	293	14.5%	266	0.752	0.936	0.988	263	0.704	0.922	0.986	268	0.106	0.526	0.879	287	0.460	0.805	0.956
0130	285	14.4%	202	0.770	0.910	0.983	193	0.714	0.911	0.969	187	0.057	0.424	0.828	279	0.347	0.549	0.835
0080	284	12.9%	141	0.757	0.935	0.987	140	0.709	0.909	0.972	278	0.035	0.278	0.720	282	0.259	0.512	0.905
0240	298	11.9%	288	0.713	0.917	0.974	276	0.599	0.837	0.922	285	0.141	0.536	0.867	290	0.367	0.709	0.923
0007	290	11.7%	284	0.791	0.897	0.948	172	0.716	0.906	0.973	277	0.069	0.591	0.902	289	0.659	0.881	0.974
Mean	379	26.1%	327	0.660	0.835	0.927	307	0.598	0.805	0.910	347	0.157	0.518	0.805	352	0.471	0.716	0.852

Table 11: 1DSfM experiment (AUC). For each scene, we list the number of input images (N_c) and the fraction of outliers. For each model, we report the number of registered cameras (N_r) and the AUC values at different error thresholds (in degrees). Winning results are marked in **bold and underlined**.

Scene	N_c	Outliers%	Ours				RESFM				Theia				GLOMAP			
			N_r	@1	@5	@30	N_r	@1	@5	@30	N_r	@1	@5	@30	N_r	@1	@5	@30
Alamo	573	32.6%	523	0.430	0.671	0.841	496	0.409	0.668	0.784	549	0.003	0.093	0.499	557	0.092	0.482	0.833
Ellis Island	227	25.1%	215	0.421	0.841	0.964	211	0.285	0.766	0.927	213	0.000	0.023	0.441	219	0.071	0.539	0.901
Madrid Metropolis	333	39.4%	298	0.590	0.819	0.944	247	0.342	0.560	0.701	319	0.015	0.241	0.650	320	0.163	0.626	0.876
Montreal Notre Dame	448	31.7%	442	0.503	0.787	0.894	333	0.528	0.790	0.933	420	0.001	0.093	0.553	444	0.090	0.549	0.890
NYC Library	330	33.6%	295	0.580	0.852	0.959	281	0.286	0.521	0.620	313	0.006	0.161	0.566	323	0.141	0.634	0.910
Notre Dame	549	35.6%	527	0.426	0.775	0.951	510	0.378	0.729	0.918	531	0.015	0.293	0.726	543	0.101	0.566	0.864
Piazza del Popolo	336	33.1%	318	0.435	0.602	0.757	248	0.533	0.802	0.893	324	0.025	0.226	0.608	331	0.203	0.648	0.899
Tower of London	467	27.0%	457	0.487	0.820	0.945	116	0.237	0.568	0.713	448	0.002	0.093	0.474	466	0.114	0.600	0.897
Vienna Cathedral	824	31.4%	763	0.371	0.586	0.709	578	0.340	0.584	0.706	767	0.000	0.008	0.270	822	0.053	0.499	0.846
Yorkminster	432	29.0%	402	0.512	0.849	0.962	304	0.080	0.404	0.628	387	0.000	0.038	0.357	418	0.103	0.600	0.905
Mean	451	31.9%	424	0.476	0.760	0.893	332	0.348	0.639	0.782	427	0.007	0.127	0.514	444	0.113	0.574	0.882

Table 12: MegaDepth experiment (registration-aware AUC). For each scene, we list the number of input images (N_c) and the fraction of outliers. For each model, we report the number of registered cameras (N_r) and the registration-aware AUC (RA-AUC) values at different error thresholds (in degrees). RA-AUC is computed over all image pairs; pairs involving at least one unregistered camera are assigned an error of 180° and therefore contribute zero recall at all reported thresholds. Winning results are marked in **bold and underlined**.

Scene	N_c	Outliers%	Ours			RESFM			Theia			GLOMAP		
			N_r	@1	@5	@30	N_r	@1	@5	@30	N_r	@1	@5	@30
0238	522	44.6%	486	0.370	0.562	0.769	354	0.251	0.373	0.421	506	0.059	0.466	0.810
0060	528	41.6%	518	0.780	0.900	0.945	491	0.659	0.783	0.838	525	0.296	0.694	0.901
0197	870	40.7%	718	0.268	0.525	0.643	706	0.423	0.570	0.628	855	0.083	0.487	0.842
0094	763	40.1%	659	0.525	0.656	0.708	539	0.182	0.266	0.374	742	0.307	0.669	0.855
0265	571	38.8%	372	0.000	0.002	0.077	457	0.083	0.410	0.583	554	0.000	0.005	0.251
0083	635	31.3%	622	0.817	0.907	0.941	572	0.694	0.778	0.802	632	0.499	0.810	0.939
0076	558	30.5%	547	0.735	0.889	0.942	512	0.616	0.784	0.830	549	0.129	0.580	0.869
0185	368	30.0%	364	0.829	0.933	0.964	345	0.714	0.840	0.872	365	0.280	0.730	0.922
0048	512	24.2%	501	0.802	0.914	0.949	485	0.718	0.844	0.876	507	0.389	0.769	0.930
0024	356	23.0%	328	0.435	0.636	0.702	308	0.391	0.620	0.709	355	0.152	0.552	0.868
0223	214	17.0%	211	0.582	0.804	0.890	202	0.430	0.673	0.770	212	0.014	0.335	0.768
5016	28	16.9%	28	0.790	0.928	0.984	28	0.355	0.499	0.741	28	0.413	0.793	0.952
0046	440	14.6%	438	0.920	0.962	0.976	388	0.472	0.534	0.573	434	0.515	0.837	0.939
0099	299	47.4%	157	0.066	0.175	0.239	222	0.026	0.176	0.350	297	0.011	0.169	0.610
1001	285	43.9%	268	0.016	0.278	0.616	256	0.004	0.194	0.628	275	0.000	0.001	0.047
0231	296	42.2%	260	0.462	0.669	0.747	247	0.377	0.595	0.664	286	0.059	0.436	0.785
0411	299	29.9%	268	0.556	0.731	0.785	273	0.578	0.764	0.820	293	0.181	0.577	0.866
0377	295	27.5%	232	0.490	0.583	0.609	197	0.342	0.418	0.440	269	0.165	0.495	0.734
0102	299	25.8%	296	0.748	0.904	0.959	280	0.595	0.794	0.857	294	0.163	0.459	0.759
0147	298	24.6%	289	0.597	0.732	0.869	241	0.272	0.361	0.454	284	0.050	0.425	0.700
0148	287	24.6%	244	0.364	0.504	0.669	256	0.331	0.434	0.534	275	0.045	0.257	0.458
0446	298	22.1%	294	0.668	0.884	0.959	288	0.586	0.832	0.909	289	0.050	0.433	0.799
0022	297	21.2%	280	0.627	0.822	0.877	273	0.571	0.774	0.832	296	0.193	0.627	0.894
0327	298	21.0%	280	0.677	0.814	0.860	270	0.616	0.764	0.807	288	0.038	0.503	0.826
0015	284	20.6%	241	0.552	0.645	0.684	223	0.337	0.442	0.498	244	0.099	0.353	0.578
0455	298	19.8%	292	0.706	0.856	0.923	293	0.688	0.885	0.946	294	0.204	0.627	0.880
0496	297	19.2%	280	0.641	0.808	0.858	281	0.550	0.784	0.862	285	0.074	0.458	0.781
1589	299	17.4%	294	0.632	0.868	0.939	290	0.700	0.875	0.928	288	0.146	0.454	0.740
0012	299	16.3%	295	0.765	0.883	0.931	291	0.646	0.781	0.816	129	0.016	0.092	0.156
0104	284	16.2%	237	0.411	0.486	0.589	195	0.342	0.439	0.465	265	0.110	0.374	0.559
0019	299	15.4%	288	0.793	0.881	0.916	247	0.567	0.656	0.677	271	0.200	0.533	0.729
0063	293	14.5%	266	0.620	0.771	0.814	263	0.567	0.743	0.794	268	0.088	0.440	0.735
0130	285	14.4%	202	0.386	0.457	0.493	193	0.327	0.417	0.444	187	0.025	0.182	0.356
0080	284	12.9%	141	0.186	0.230	0.242	140	0.172	0.220	0.235	278	0.033	0.267	0.690
0240	298	11.9%	288	0.666	0.856	0.910	276	0.514	0.718	0.791	285	0.129	0.490	0.792
0007	290	11.7%	284	0.759	0.860	0.909	172	0.251	0.318	0.342	277	0.063	0.539	0.823
Mean	379	26.1%	327	0.562	0.703	0.775	307	0.443	0.593	0.670	347	0.147	0.470	0.726

Table 13: 1DSfM experiment (registration-aware AUC). For each scene, we list the number of input images (N_c) and the fraction of outliers. For each model, we report the number of registered cameras (N_r) and the registration-aware AUC (RA-AUC) values at different error thresholds (in degrees). RA-AUC is computed over all image pairs; pairs involving at least one unregistered camera are assigned an error of 180° and therefore contribute zero recall at all reported thresholds. Winning results are marked in **bold and underlined**.

Scene	N_c	Outliers%	Ours			RESFM			Theia			GLOMAP		
			N_r	@1	@5	@30	N_r	@1	@5	@30	N_r	@1	@5	@30
Alamo	573	32.6%	523	0.358	0.559	0.701	496	0.306	0.500	0.587	549	0.002	0.085	0.458
Ellis Island	227	25.1%	215	0.378	0.754	0.865	211	0.246	0.662	0.801	213	0.000	0.020	0.388
Madrid Metropolis	333	39.4%	298	0.472	0.656	0.756	247	0.188	0.308	0.385	319	0.013	0.221	0.596
Montreal Notre Dame	448	31.7%	442	0.490	0.766	0.870	333	0.291	0.436	0.515	420	0.001	0.082	0.486
NYC Library	330	33.6%	295	0.463	0.681	0.766	281	0.207	0.378	0.449	313	0.005	0.145	0.509
Notre Dame	549	35.6%	527	0.393	0.714	0.876	510	0.326	0.629	0.792	531	0.014	0.274	0.679
Piazza del Popolo	336	33.1%	318	0.390	0.539	0.678	248	0.323	0.436	0.486	324	0.023	0.210	0.565
Tower of London	467	27.0%	457	0.466	0.785	0.905	116	0.015	0.035	0.044	448	0.002	0.085	0.436
Vienna Cathedral	824	31.4%	763	0.318	0.502	0.608	578	0.167	0.287	0.347	767	0.000	0.007	0.234
Yorkminster	432	29.0%	402	0.443	0.735	0.833	304	0.040	0.200	0.311	387	0.000	0.030	0.287
Mean	451	31.9%	424	0.417	0.669	0.786	332	0.211	0.387	0.472	427	0.006	0.116	0.464