VGPA: DEEP VIEW-GRAPH POSE AVERAGING FOR STRUCTURE-FROM-MOTION

Anonymous authors

Paper under double-blind review

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. A fast view re-integration step increases camera coverage by reintroducing discarded images. 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.

1 Introduction

Camera pose recovery is an essential part of 3D scene reconstruction and view synthesis applications. Many common Multiview Stereo (MVS) (Seitz et al., 2006; Yao et al., 2018) and view synthesis methods, including Neural Radiance Fields (NeRF) (Mildenhall et al., 2021) and Gaussian Splatting (GS) (Kerbl et al., 2023) 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 (Schönberger & Frahm, 2016), 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 (Sturm & Triggs, 1996) simultaneously solve for all cameras and point tracks. These methods, however, attempt to factor large tensors that include all the track points.

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 (Moulon et al., 2016; Sweeney et al., 2015), while some recent works developed techniques for directly averaging essential and fundamental matrices (Kasten et al., 2019a;b). 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

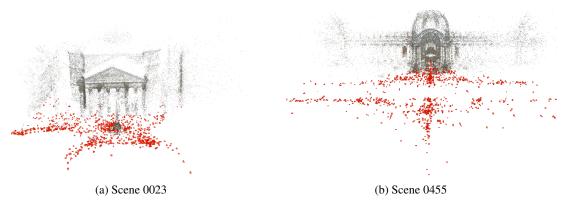


Figure 1: 3D reconstructions and recovered camera parameters produced by VGPA on two large scenes ($N_c > 1000$ images). VGPA registers almost all images and scales to thousand-image collections, in contrast to existing image-based deep methods (e.g., VGGT, VGGSfM).

deep-based approaches to SfM (Khatib et al., 2025; Moran et al., 2021; Brynte et al., 2023), 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 efficient *view reintegration* step is applied to recover cameras that were discarded in the process by the network, increasing camera coverage.

Our approach is efficient and achieves high accuracy. It copes well with large-scale inputs, including ones with more than a thousand images. Moreover, our method is agnostic to the density of the graph. We obtain comparable performance when constructing the view graph using top-k neighbors retrieved with NetVLAD (Arandjelovic et al., 2016), instead of exhaustive pairwise matching, despite the large difference in edge density. We note that in the uncalibrated 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 (Khatib et al., 2025; Moran et al., 2021; Brynte et al., 2023), 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 (Khatib et al., 2025; Moran et al., 2021; Brynte et al., 2023) often struggle.

Below we summarize our contributions.

- 1. **VGPA:** an efficient deep, permutation-equivariant GNN for *view-graph pose averaging* that predicts global camera extrinsics from noisy pairwise estimates.
- 2. **Unsupervised training by only enforcing relative-pose consistency**, without reprojection supervision or 3D point prediction; structure is recovered by triangulation + robust BA.
- 3. **Robustness to graph density,** achieving similar accuracy with both exhaustive SIFT matching and top-20 NetVLAD view graphs, despite large differences in edge density.
- 4. **Works with unknown intrinsics.** We show VGPA remains accurate when intrinsics are initialized simply and optimized during BA.
- View re-integration. A lightweight add-back loop improves coverage with small runtime overhead.

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. (Agarwal et al., 2011; Schönberger & Frahm, 2016; Snavely et al., 2006; Wu, 2013). 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 (Sturm & Triggs, 1996; Dai et al., 2010; Lin et al., 2017). 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 (Moran et al., 2021; Brynte et al., 2023; Chen et al., 2024; Khatib et al., 2025). 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 (Khatib et al., 2025). 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 (Martinec & Pajdla, 2007; Özyeşil et al., 2017; Sweeney et al., 2015; Moulon et al., 2016). Kasten et al. (2019a;b) introduced an averaging method for averaging essential and fundamental matrices, solving for all of these parameters in a single optimization. With the exception of (Pan et al., 2024), these methods require a separate step of 3D point triangulation. Theia (Sweeney et al., 2015) and the recent GLOMAP (Pan et al., 2024), in particular, were shown to yield accurate recovery.

Several recent works train networks to solve rotation averaging on the view graph. NeurORA (Purkait et al., 2020) learns to denoise pairwise relative rotations and aggregates them to recover global orientations, while (Li & Ling, 2021) 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 (Wang et al., 2023a), DUST3R (Wang et al., 2023b), and MAST3R (Leroy et al., 2024) are restricted to processing only a small number of input images, whereas Ace-Zero (Brachmann et al., 2024) and FlowMap (Smith et al., 2024) are tailored for video sequences under constant illumination. More recently, VGGT (Wang et al., 2025) 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 \sim 200. Fast3R (Yang et al., 2025) scales to larger collections but typically attains lower accuracy than VGGT at comparable settings.

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. It remains robust to heavy outlier contamination in realistic point track data.

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 $I_1, ..., I_m$. Following standard SfM pipelines, we begin by detecting and matching features across the images using standard algorithms such as SIFT or SuperPoint (DeTone et al., 2018; Lowe, 2004). We next apply RANSAC (Bolles & Fischler, 1981) and obtain a partial

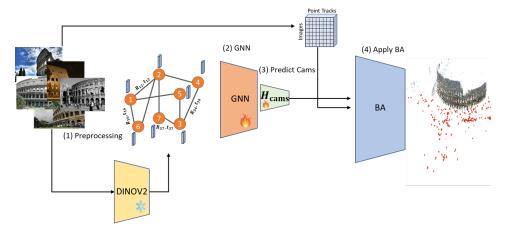


Figure 2: **Method overview.** (1) *Preprocessing:* estimate pairwise relative poses from essential matrices and build the view graph; extract point tracks and frozen DINOv2 image descriptors. (2) *GNN:* a permutation-equivariant, edge-conditioned GNN aggregates the view graph to produce camera embeddings. (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.

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., (Schönberger & Frahm, 2016)) to join such pairs to form longer tracks. Each track is a set $T_k = \{\mathbf{x}_{i_1,k},\mathbf{x}_{i_2,k},...\}$ with $i_1,i_2,...\in[m]$, and we assume that T_k contains the projected locations of a single 3D scene point, denoted \mathbf{X}_k , onto $I_{i_1},I_{i_2},....$ 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

$$\mathbf{e}_{ij}^{(0)} = \phi_{\textit{e}}\!\!\left([\log R_{ij}^{\text{RANSAC}},\, \hat{\mathbf{t}}_{ij}^{\text{RANSAC}}]\right)\!,$$

where $\log: SO(3) \to \mathfrak{so}(3)$ is the matrix logarithm and $\hat{\mathbf{t}}_{ij}^{\text{RANSAC}} \in \mathbb{S}^2$ is the *unit normed*-translation direction recovered from the essential matrix. To inject image-level context, each node v_i is initialized with the DINOv2 [CLS] token $\mathbf{h}_i^{(0)}$ computed from image I_i .

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

$$\begin{split} \tilde{\mathbf{m}}_{i}^{(\ell)} &= \sum_{j \in \mathcal{N}(i)} \phi_{m} \big(\mathbf{h}_{i}^{(\ell)}, \, \mathbf{h}_{j}^{(\ell)}, \, \mathbf{e}_{ij}^{(\ell)} \big), \\ \mathbf{m}_{i}^{(\ell)} &= \frac{1}{|\mathcal{N}(i)|} \, \tilde{\mathbf{m}}_{i}^{(\ell)}, \\ \mathbf{h}_{i}^{(\ell+1)} &= \mathrm{LN} \Big(\mathbf{h}_{i}^{(\ell)} + \mathrm{Drop} \big(\psi \big(\mathrm{LN}(\mathbf{h}_{i}^{(\ell)}), \, \mathbf{m}_{i}^{(\ell)} \big) \big) \Big), \end{split}$$

for $\ell=0,\ldots,L-1$, where ϕ_m and ψ are MLPs, LN denotes Layer Normalization (Ba et al., 2016), and Drop denotes Dropout (Srivastava et al., 2014). The network is equivariant to node relabelings (permutations) of \mathcal{G} . 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,

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

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,\ldots,P_m . Each camera is parameterized as $P_i=[R_i\mid \mathbf{t}_i]$ with $R_i\in SO(3)$ 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

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

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

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

 R_{ij}^{RANSAC} and $\mathbf{t}_{ij}^{\mathrm{RANSAC}}$ are the corresponding rotation and translation obtained with RANSAC in preprocessing, $d_R(R_1,R_2) = \arccos\left(\frac{\mathrm{trace}(R_1^{\top}R_2)-1}{2}\right)$ 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. For each scene, we sample uniformly $s \in [0.1, 0.2]$ of the images (without replacement) to form a subgraph. Models are selected by early stopping on a held-out set; we report the checkpoint with the lowest validation error. Additional details appear in the Appendix.

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 $T_{\rm ft} = 200$ steps. Next, we triangulate using DLT (Hartley & Zisserman, 2003) 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 (Li & Snavely, 2018) 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 (Wilson & Snavely, 2014), Strecha (Strecha et al., 2008), and BlendedMVS (Yao et al., 2020). We refer the reader to the supplementary material for hyperparameters and further technical details.

MegaDepth (Li & Snavely, 2018). The MegaDepth dataset includes 196 different outdoor landmark scenes curated from the internet. We followed the train/test split as in (Khatib et al., 2025), 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 (Wilson & Snavely, 2014). 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 (Strecha et al., 2008). The Strecha dataset consists of five small outdoor scenes (≤ 30 images) and includes ground-truth data acquired with a LIDAR system. We test our method on four of these five scenes.

BlendedMVS (Yao et al., 2020). 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 measurement, and, therefore, as is common in the field, we use camera poses computed with COLMAP Schönberger & Frahm (2016), 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 Jiang et al. (2013); Wilson & Snavely (2014); Cui & Tan (2015); Ozyesil & Singer (2015); Brynte et al. (2023); Khatib et al. (2025); Zhang et al. (2024)) due to its accurate and robust performance. To evaluate our method with real ground truth, we additionally show results on the smaller datasets Strecha (Strecha et al., 2008) and BlendedMVS (Yao et al., 2020).

4.2 BASELINES

With the exception of VGGT (Wang et al., 2025), the settings and results for all baselines below were taken from (Khatib et al., 2025).

RESfM (**Khatib et al., 2025**). 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.

VGGSfM (Wang et al., 2024) is a differentiable, trainable SfM pipeline.

MASt3R (Leroy et al., 2024). An SfM pipeline that utilizes a global alignment procedure to merge pairwise pointmap predictions.

Theia (Sweeney et al., 2015). A global SfM pipeline that applies rotation averaging, followed by translation averaging, and finally 3D point triangulation.

GLOMAP (Pan et al., 2024). A global SfM pipeline that first applies rotation averaging, followed by an integrated step of translation averaging and point triangulation.

VGGT (Wang et al., 2025). 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.

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 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 the baseline methods (except VGGSfM, VGGT and MAST3R, which are applied directly to the input images) were run with the same set of point tracks. For all methods, we apply a final post-processing step of robust bundle adjustment.

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. Compared 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 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 (in degrees) and translation errors. (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. Yellow represents the best result among the deep-based algorithms and green among the classical algorithms.

C	3.7	0-41:01	ı	Ours			RESf	M		Theia			GLOMA	AΡ
Scene	N_c	Outliers%	N_r	Rot	Trans	N_r	Rot	Trans	$ N_r $	Rot	Trans	N_r	Rot	Trans
0238	522	44.6%	488	4.50	0.686	283	2.61	0.325	506	1.21	0.334	499	0.74	0.349
0060	528	41.6%	518	0.07	0.014	503	0.29	0.029	$\overline{525}$	0.85	0.124	522	0.11	0.048
0197	870	40.7%	641	1.28	0.271	667	4.22	0.333	855	1.16	0.227	814	0.43	0.129
0094	763	40.1%	663	0.66	0.101	537	3.77	0.750	$\overline{742}$	0.75	0.160	717	0.88	3.907
0265	571	38.8%	345	2.93	0.998	346	1.25	0.389	554	5.83	2.216	558	7.46	2.839
0083	635	31.3%	614	0.06	0.005	596	0.64	0.058	632	0.37	0.372	614	0.08	0.016
0076	558	30.5%	543	$\overline{0.09}$	0.016	524	0.37	0.094	$\overline{549}$	0.78	0.120	541	0.17	0.042
0185	368	30.0%	358	0.10	0.022	350	0.06	0.010	365	0.41	0.094	365	0.16	0.051
0048	512	24.2%	500	0.29	0.026	474	4.69	0.178	507	0.41	0.105	506	0.15	0.224
0024	356	23.0%	313	3.38	0.772	309	2.03	0.398	355	0.56	0.219	339	0.15	0.104
0223	214	17.0%	208	2.75	0.195	204	3.76	0.510	212	3.34	0.519	214	1.75	0.275
5016	28	16.9%	28	0.08	$\overline{0.015}$	28	0.12	0.016	28	0.10	0.061	28	$\overline{0.08}$	0.046
0046	440	14.6%	$4\overline{39}$	0.54	0.071	399	0.95	0.043	434	0.25	0.112	$\overline{440}$	$\overline{0.03}$	0.007
1001	285	43.9%	265	1.89	3.840	251	1.70	0.661	276	7.97	4.014	281	4.56	3.817
0231	296	42.2%	261	0.24	0.030	246	0.84	0.065	286	1.37	0.322	279	0.73	0.134
0411	299	29.9%	270	$\overline{0.12}$	0.018	273	0.13	0.020	293	0.39	0.196	269	0.19	0.148
0377	295	27.5%	232	0.30	0.035	210	0.29	0.018	269	1.13	0.205	268	0.65	0.237
0102	299	25.8%	297	0.18	0.031	284	0.28	0.059	294	2.31	0.698	293	0.15	0.101
0147	298	24.6%	282	1.99	0.153	207	4.62	0.325	284	6.36	0.934	290	6.75	3.542
0148	287	24.6%	211	0.93	0.037	197	0.60	0.035	275	13.98	1.558	283	22.73	2.646
0446	298	22.1%	292	0.22	0.019	288	0.72	0.046	289	1.23	0.391	296	0.20	0.071
0022	297	21.2%	277	0.29	0.044	274	0.29	0.039	296	0.58	0.160	281	0.22	0.087
0327	298	21.0%	<u>291</u>	0.12	0.014	271	0.26	0.090	288	1.27	0.360	290	15.54	2.035
0015	284	20.6%	243	0.52	0.058	215	1.04	0.167	244	2.21	0.389	274	0.28	0.095
0455	298	19.8%	290	0.39	0.078	293	0.68	0.105	294	0.77	0.159	298	0.35	0.064
0496	297	19.2%	279	0.37	0.033	281	0.35	0.055	285	1.40	0.550	291	0.44	0.303
1589	299	17.4%	296	0.11	0.010	290	0.14	0.019	288	0.82	0.193	299	0.07	0.041
0012	299	16.3%	<u>295</u>	0.63	0.071	287	0.40	0.027	129	1.04	0.318	295	0.51	0.121
0019	299	15.4%	291	0.37	0.020	250	0.06	0.008	271	0.81	0.250	296	0.09	0.025
0063	293	14.5%	268	0.18	0.025	262	0.46	0.048	268	0.92	0.605	288	0.32	0.100
0130	285	14.4%	199	5.12	0.618	192	0.20	0.023	187	1.20	0.349	<u>281</u>	2.00	0.909
0080	284	12.9%	162	0.58	0.109	139	0.59	0.096	278	2.62	0.868	<u>283</u>	1.92	0.237
0240	298	11.9%	295	0.64	3.479	275	3.13	0.265	278	1.31	0.470	294	0.39	0.135
0007	290	11.7%	283	1.53	0.150	172	0.91	0.041	277	1.24	0.174	<u>290</u>	0.19	0.035

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 (N_r) and mean values of the rotation (in degrees) and translation errors. Winning results are marked in bold and underlined. Yellow represents the best result among the deep-based algorithms and green among the classical algorithms.

Scene	N.T	Outliers%	% Ours			RESFM		Theia			GLON		1AP	
Scelle	N_c	Outriers%	N_r	Rot	Trans	N_r	Rot	Trans	N_r	Rot	Trans	N_r	Rot	Trans
Alamo	573	32.6%	509	1.50	0.342	484	3.66	0.515	553	4.42	1.433	<u>557</u>	2.45	1.520
Ellis Island	227	25.1%	214	0.27	0.077	214	0.82	0.122	213	5.01	1.527	<u>219</u>	0.58	0.155
Madrid Metropolis	333	39.4%	295	1.47	0.136	244	8.42	0.827	-	-	-	320	1.22	0.242
Montreal Notre Dame	448	31.7%	425	0.34	0.073	346	2.82	0.352	422	4.47	1.285	<u>444</u>	0.60	0.211
NYC Library	330	33.6%	285	1.20	0.422	224	3.96	0.429	<u>534</u>	4.06	1.141	323	0.58	0.189
Notre Dame	549	35.6%	519	0.64	0.065	517	1.20	0.231	314	3.70	0.828	<u>543</u>	2.73	0.389
Piazza del Popolo	336	33.1%	315	4.42	0.710	249	2.20	0.186	325	3.31	1.053	331	0.80	0.188
Tower of London	467	27.0%	454	0.78	0.073	94	0.67	0.026	448	6.61	1.189	<u>466</u>	0.81	0.138
Vienna Cathedral	824	31.4%	753	19.28	1.285	479	1.52	0.112	772	12.25	1.663	822	2.00	2.414
Yorkminster	432	29.0%	403	1.38	0.144	331	14.54	1.468	390	8.35	1.916	<u>418</u>	$\underline{0.95}$	0.316

Following Khatib et al. (2025), we evaluate VGPA on the smaller Strecha and BlendedMVS benchmarks, which provide ground-truth camera poses. As shown in Table 3, 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).

Robustness to view-graph density. We train VGPA using relative poses estimated from SIFT features with **exhaustive** mutual nearest-neighbor (MNN) matching. Here, at test time, we vary the view-graph sparsity using NetVLAD retrieval, connecting each image to its top-k nearest neighbors (default k=20). As shown in Table 4, VGPA achieves accuracy comparable to the exhaustive SIFT+MNN graph while using far fewer edges, and its performance varies only marginally across a wide range of k, provided the graph remains well connected.

Postprocessing (view re-integration). Since our pipeline may discard some of the images during the BA stage, we attempt in postprocessing to re-register discarded images using a lightweight addback loop. Unregistered views are ranked by connectivity (e.g., number of 2D-3D matches) with

Table 3: **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	37.	3t Cl	1		Ours			V	GGT			MA	St3R			VG	GSfM			T	heia			CO	LMAP			GLO	OMAP	
Scene	IV _c	Jut.%	N_r	Rot	Ours 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 R	Rot	Trans	Time
														Stre																
entry-P10	10	4.8	10	0.004	4 0.0005	10.0	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 0	0.023	0.007	36.0	10 0.	187	0.026	12.5
fountain-P11	11	1.4	11	0.012	2 0.0005	14.7	11	0.034	0.019	12.2	11 (0.160	0.026	22	11	0.172	0.016	15.4	11	0.027	0.002	1.5	110	0.027	0.003	37.0	11 0.	194	0.022	38.6
Herz-Jesus-P8	8	1.8	8	0.009	0.0010	7.4	8	0.032	0.011	12.7	8 (0.363	0.037	16	8	0.206	0.042	8.7	8	0.025	0.005	0.6	8 (0.026	0.004	22.0	8 0.	091	0.015	5.0
Herz-Jesus-P25	25	2.8	25	0.010	0.0003	12.5	25	0.048	0.007	31.9	25 (0.869	0.057	81	25	0.158	0.046	19.6	25	0.026	0.006	2.4	25 0	0.028	0.006	60.0	25 0.	138	0.013	76.6
														lende																
scene0	75	2.0	74	0.019	0.0011	136	75	0.041	0.017	108	75 C	0.501	0.191	516	75	0.045	0.0106	61	75	0.009	0.0017	49	75 0	0.006	0.0005	106	75 <u>0.</u>	007	0.0016	198
scene1	51	1.4	51	0.341	0.0342	38	51	0.101	0.050	41	51 (0.919	0.173	1017	51	0.098	0.0112	32	51	0.029	0.0099	18	51 0	0.007	0.0003	67	51 <u>0.</u>	024	0.0102	117
scene2	33	2.2	33	0.008	0.0004	19	33	0.230	0.022	52	33 1	1.972	0.130	117	33	0.227	0.0180	30	33	0.045	0.0098	15	33 0	0.003	0.0002	55	33 0.0	025 (0.0060	87
scene3	66	8.8	66	0.006	0.0065	65	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	0.004	0.0002	128	66 0.	008	0.0017	392

Table 4: **Robustness to graph density**. For each scene we list the number of input images (N_c) . Our default setting uses *Exhaustive SIFT*, and we also report results with NetVLAD@15 + SIFT. For both, we show the number of registered images (N_r) , mean rotation error (deg), and mean translation error.

Caana	A .7	Our	s (Exhaustive	SIFT, default) Trans	Our	s (NetVLAD	@15 + SIFT)
Scene	IV_C	N_r	Rot	Trans	N_r	Rot	Trans
Alamo	573	509	1.5	0.342	505	1.4	0.412
Ellis Island	227	214	0.27	0.077	216	0.36	0.172
Madrid Metropolis	333	295	1.47	0.136	290	1.6	0.244

our point cloud. For each candidate, we determine its pose using the available 2D–3D point matches and refined with a brief BA applied locally to its neighboring views. The process iterates until no further views can be added. Table 8, in the appendix, compares Ours and Ours + post-processing in terms of N_r , mean rotation error (deg), and mean translation error, showing that the add-back step increases camera coverage with minimal runtime overhead (1 sec for each added image).

Uncalibrated image collections. Table 5 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. While self-calibration incurs a small accuracy drop relative to ground-truth intrinsics, VGPA remains competitive and maintains high performance.

Table 5: **BlendedMVS: 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 (optimized) and report N_r , mean rotation error (deg), and mean translation error. Best results are in **bold**.

Caama	A .7	Out 07	Our	s (w/ int	trinsics)	Our	s (w/o in	trinsics)
Scelle	IV_C	Out.%	N_r	Rot	trinsics) Trans	$ N_r $	Rot	Trans
scene0		2.0	74	0.019	0.0011	74	0.019	0.0019
scene1	51	1.4	51	0.341	0.0342	51	0.338	0.0400
scene2	33	2.2	33	0.008	0.0004	33	0.025	0.0049
scene3	66	8.8	66	0.006	0.0065	66	0.010	0.0014

Qualitative results. Figure 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 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 are provided in the Appendix.

Runtime.

Table 6 reports wall-clock runtimes on the identical (contaminated) point tracks produced by our preprocessing. VGPA is substantially faster than COLMAP, GLOMAP, and Theia, and remains competitive in 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.

Table 6: **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.

Scene	N.T.	O41:0/	O	urs		RE	SfM		COL	MAP		Th	eia		GLO	MAP	
Scene	N_c	Outliers%	Total (Mins)	N_r	$N_r/t\uparrow$												
Alamo	573	32.6	4.4	509	116.2	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	1.1	214	194.4	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	1.7	295	172.5	5.8	244	42.1	25.1	323	12.9	-	-	-	7.1	320	45.2
Montreal Notre Dame	448	31.7	2.8	425	151.8	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	2.9	519	179.5	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	1.3	285	212.7	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	1.1	315	277.6	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	3.3	454	137.6	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	7.5	753	101.0	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	403	140.5	7.7	331	42.9	31.4	419	13.3	2.9	390	135.3	14.8	418	28.2
Mean	-	-	2.9	417	168.4	9.8	318	45.3	46.4	448	13.1	5.7	441	117.2	18.2	444	31.5

Ablations

Ablations confirm that each core component of our method is critical. Removing subset sampling substantially increases both rotation and translation errors, showing its importance for robustness. Excluding DINO appearance cues or reducing the number of GNN layers also leads to a modest decline. Most importantly, fine-tuning yields a large improvement, reducing both rotation and translation errors to their lowest values. See Table 7, where we report errors *before* the final BA refinement.

Table 7: Ablation study reporting mean rotation and translation errors *before* final BA refinement. Fine-tuning yields the lowest errors.

	Mean Rotation Error (↓)	Mean Translation Error (↓)
Ours w/o subset sampling	12.9	2.5
Ours w/o image features	9.8	2.2
Ours w/ 2 layers	10.1	2.2
Ours (base model)	9.5	2.1
Proposed (with fine-tuning)	1.9	0.5

5 CONCLUSION

We present VGPA, a joint, unsupervised deep pose averaging network for multiview SfM. The design includes a pose-averaging GNN that enforces the consistency of relative pairwise rotations and translation directions while also considering image-level context. Additional 3D point triangulation and robust BA refinement ensure high accuracy and 3D structure recovery. Across challenging benchmarks (including MegaDepth, 1DSfM), our VGPA method outperforms deep methods and remains competitive with strong classical pipelines while maintaining a high camera coverage. A lightweight post-processing sweep can reintroduce the few remaining views with negligible overhead.

REFERENCES

- Sameer Agarwal, Keir Mierle, and Others. Ceres solver. http://ceres-solver.org.
 - Sameer Agarwal, Yasutaka Furukawa, Noah Snavely, Ian Simon, Brian Curless, Steven M Seitz, and Richard Szeliski. Building rome in a day. *Communications of the ACM*, 54(10):105–112, 2011.
 - Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. Netvlad: Cnn architecture for weakly supervised place recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5297–5307, 2016.
 - Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint* arXiv:1607.06450, 2016.
 - Robert C Bolles and Martin A Fischler. A ransac-based approach to model fitting and its application to finding cylinders in range data. In *IJCAI*, volume 1981, pp. 637–643, 1981.
 - Eric Brachmann, Jamie Wynn, Shuai Chen, Tommaso Cavallari, Áron Monszpart, Daniyar Turmukhambetov, and Victor Adrian Prisacariu. Scene coordinate reconstruction: Posing of image collections via incremental learning of a relocalizer. *arXiv* preprint arXiv:2404.14351, 2024.
 - Lucas Brynte, José Pedro Iglesias, Carl Olsson, and Fredrik Kahl. Learning structure-from-motion with graph attention networks. *arXiv preprint arXiv:2308.15984*, 2023.
 - Zequan Chen, Jianping Li, Qusheng Li, Bisheng Yang, and Zhen Dong. Deepaat: Deep automated aerial triangulation for fast uav-based mapping, 2024.
 - Zhaopeng Cui and Ping Tan. Global structure-from-motion by similarity averaging. In *Proceedings* of the IEEE International Conference on Computer Vision, pp. 864–872, 2015.
 - Yuchao Dai, Hongdong li, and Mingyi He. Element-wise factorization for n-view projective reconstruction. pp. 396–409, 09 2010. ISBN 978-3-642-15560-4. doi: 10.1007/978-3-642-15561-1_29.
 - Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pp. 224–236, 2018.
 - Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.
 - Nianjuan Jiang, Zhaopeng Cui, and Ping Tan. A global linear method for camera pose registration. In *Proceedings of the IEEE international conference on computer vision*, pp. 481–488, 2013.
 - Yoni Kasten, Amnon Geifman, Meirav Galun, and Ronen Basri. 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, 2019a.
 - Yoni Kasten, Amnon Geifman, Meirav Galun, and Ronen Basri. 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, 2019b.
 - Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023.
 - Fadi Khatib, Yoni Kasten, Dror Moran, Meirav Galun, and Ronen Basri. Resfm: Robust deep equivariant structure from motion. In *The Thirteenth International Conference on Learning Representations*, 2025.
 - Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
 - Vincent Leroy, Yohann Cabon, and Jérôme Revaud. Grounding image matching in 3d with mast3r. *arXiv preprint arXiv:2406.09756*, 2024.

- Xinyi Li and Haibin Ling. Pogo-net: Pose graph optimization with graph neural networks. In Proceedings of the IEEE/CVF international conference on computer vision, pp. 5895–5905, 2021.
- 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*, pp. 2041–2050, 2018.
 - Yang Lin, Li Yang, Zhouchen Lin, Tong Lin, and Hongbin Zha. Factorization for projective and metric reconstruction via truncated nuclear norm. In 2017 International Joint Conference on Neural Networks (IJCNN), pp. 470–477, 2017. doi: 10.1109/IJCNN.2017.7965891.
 - David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.
 - Daniel Martinec and Tomas Pajdla. Robust rotation and translation estimation in multiview reconstruction. In 2007 IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–8. IEEE, 2007.
 - Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021.
 - Dror Moran, Hodaya Koslowsky, Yoni Kasten, Haggai Maron, Meirav Galun, and Ronen Basri. Deep permutation equivariant structure from motion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5976–5986, 2021.
 - Pierre Moulon, Pascal Monasse, Romuald Perrot, and Renaud Marlet. Openmyg: Open multiple view geometry. In *International Workshop on Reproducible Research in Pattern Recognition*, pp. 60–74. Springer, 2016.
 - Onur Ozyesil and Amit Singer. Robust camera location estimation by convex programming. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2674–2683, 2015.
 - Onur Özyeşil, Vladislav Voroninski, Ronen Basri, and Amit Singer. A survey of structure from motion*. *Acta Numerica*, 26:305–364, 2017.
 - Linfei Pan, Dániel Baráth, Marc Pollefeys, and Johannes L Schönberger. Global structure-frommotion revisited. In *European Conference on Computer Vision (ECCV)*, 2024.
 - Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.
 - Pulak Purkait, Tat-Jun Chin, and Ian Reid. Neurora: Neural robust rotation averaging. In *European conference on computer vision*, pp. 137–154. Springer, 2020.
 - Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
 - Steven M Seitz, Brian Curless, James Diebel, Daniel Scharstein, and Richard Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In 2006 IEEE computer society conference on computer vision and pattern recognition (CVPR'06), volume 1, pp. 519–528. IEEE, 2006.
 - Cameron Smith, David Charatan, Ayush Tewari, and Vincent Sitzmann. Flowmap: High-quality camera poses, intrinsics, and depth via gradient descent. arXiv preprint arXiv:2404.15259, 2024.
 - Noah Snavely, Steven M Seitz, and Richard Szeliski. Photo tourism: exploring photo collections in 3d. In *ACM siggraph 2006 papers*, pp. 835–846. 2006.
 - Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

- Christoph Strecha, Wolfgang Von Hansen, Luc Van Gool, Pascal Fua, and Ulrich Thoennessen. 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.
- Peter Sturm and Bill Triggs. A factorization-based algorithm for multi-image projective structure and motion. In *European conference on computer vision*, pp. 709–720. Springer, 1996.
- Christopher Sweeney, Tobias Hollerer, and Matthew Turk. Theia: A fast and scalable structure-from-motion library. In *Proceedings of the 23rd ACM international conference on Multimedia*, pp. 693–696, 2015.
- Jianyuan Wang, Nikita Karaev, Christian Rupprecht, and David Novotny. Visual geometry grounded deep structure from motion. *arXiv preprint arXiv:2312.04563*, 2023a.
- Jianyuan Wang, Nikita Karaev, Christian Rupprecht, and David Novotny. 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.
- Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 5294–5306, 2025.
- Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. *arXiv preprint arXiv:2312.14132*, 2023b.
- Kyle Wilson and Noah Snavely. 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.
- Changchang Wu. Towards linear-time incremental structure from motion. In 2013 International Conference on 3D Vision-3DV 2013, pp. 127–134. IEEE, 2013.
- Jianing Yang, Alexander Sax, Kevin J Liang, Mikael Henaff, Hao Tang, Ang Cao, Joyce Chai, Franziska Meier, and Matt Feiszli. 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.
- Yao Yao, Zixin Luo, Shiwei Li, Tian Fang, and Long Quan. Mvsnet: Depth inference for unstructured multi-view stereo. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 767–783, 2018.
- Yao Yao, Zixin Luo, Shiwei Li, Jingyang Zhang, Yufan Ren, Lei Zhou, Tian Fang, and Long Quan. Blendedmys: 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.
- Jason Y Zhang, Amy Lin, Moneish Kumar, Tzu-Hsuan Yang, Deva Ramanan, and Shubham Tulsiani. Cameras as rays: Pose estimation via ray diffusion. In *International Conference on Learning Representations (ICLR)*, 2024.

APPENDIX

A QUALITATIVE RESULTS

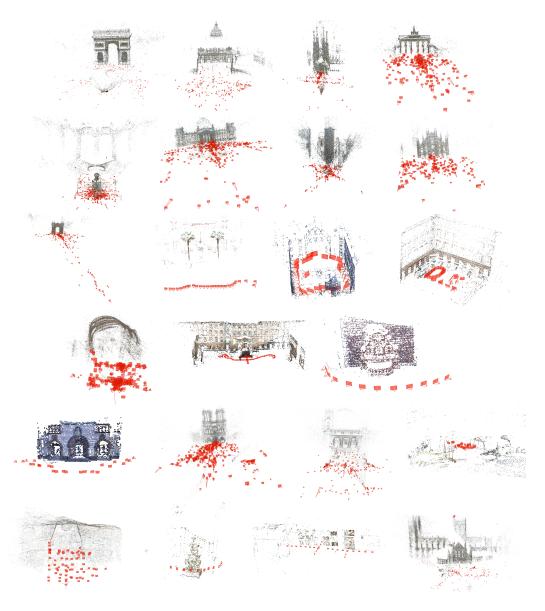


Figure 3: Example reconstructions from the proposed VGPA on various datasets.

USE OF LARGE LANGUAGE MODELS (LLMS) We used a large language model (ChatGPT) solely for language polishing, i.e., improving grammar, clarity, and style of sentences. 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 (Paszke et al., 2019) and the Adam optimizer (Kingma & Ba, 2014) with gradient normal-ization.

Training. Each epoch iterates over all training scenes. For every scene, we uniformly sample (without replacement) 10%–20% of the images to form the training subgraph. A held-out validation set is used for early stopping. Validation and test evaluations use the complete view graph. Training on MegaDepth takes roughly 8 hours on two A100s. We fix the random seed to 20.

Architecture details. The encoder uses 3 edge-conditioned message passing layers with 256 channels (nodes and edges) and ReLU activations. The camera head $H_{\rm cams}$ is a 3-layer MLP with 256 channels.

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 Agarwal et al. with a Huber loss (scale 0.1) for robustness, following Khatib et al. (2025). In each BA round, we cap the number of iterations at 300 or stop earlier on convergence.

C CONSTRUCTING POINT TRACKS

We follow the preprocessing in Khatib et al. (2025) to construct point tracks; see their Appendix for full details.

D ADDITIONAL RESULTS

Here we present the view re-integration results (Table 8), while Tables 9 and 10 report the median results for the MegaDepth and 1DSfM experiments.

Table 8: **MegaDepth: Effect of View Re-Integration.** 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 step (Ours + Post-Processing).

Scene	N_c	Outliers%	I	Ours				rocessing
Scene	1 V C	Outliers 70	N_r	Rot	Trans	$ N_r $	Rot	Trans
0238	522	44.6%	488	4.50	0.686	511	4.43	0.681
0060	528	41.6%	518	0.07	0.014	526	0.08	0.016
0197	870	40.7%	641	1.28	0.271	749	1.33	0.291
0094	763	40.1%	663	0.66	0.101	708	1.24	0.134
0265	571	38.8%	345	2.93	0.998	476	3.50	1.077
0083	635	31.3%	614	0.06	0.005	628	0.07	0.007
0076	558	30.5%	543	0.09	0.016	553	0.11	0.018
0185	368	30.0%	358	0.10	0.022	364	0.11	0.022
0048	512	24.2%	500	0.29	0.026	507	0.29	0.026
0024	356	23.0%	313	3.38	0.772	342	3.39	0.781
0223	214	17.0%	208	2.75	0.195	213	3.56	0.289
5016	28	16.9%	28	0.08	0.015	28	0.08	0.015
0046	440	14.6%	439	0.54	0.071	440	0.54	0.071
1001	285	43.9%	265	1.89	3.840	274	1.86	3.938
0231	296	42.2%	261	0.24	0.030	271	0.45	0.061
0411	299	29.9%	270	0.12	0.018	289	0.13	0.021
0377	295	27.5%	232	0.30	0.035	253	0.32	0.036
0102	299	25.8%	297	0.18	0.031	299	0.18	0.031
0148	287	24.6%	211	0.93	0.037	225	2.28	0.229
0147	298	24.6%	282	1.99	0.153	292	1.98	0.153
0446	298	22.1%	292	0.22	0.019	297	0.24	0.021
0022	297	21.2%	277	0.29	0.044	287	0.29	0.044
0327	298	21.0%	291	0.12	0.014	293	0.12	0.014
0015	284	20.6%	243	0.52	0.058	255	0.68	0.111
0455	298	19.8%	290	0.39	0.078	298	0.52	0.092
0496	297	19.2%	279	0.37	0.033	290	0.38	0.035
1589	299	17.4%	296	0.11	0.010	298	0.11	0.010
0012	299	16.3%	295	0.63	0.071	298	1.04	0.122
0019	299	15.4%	291	0.37	0.020	297	0.52	0.026
0063	293	14.5%	268	0.18	0.025	274	0.20	0.026
0130	285	14.4%	199	5.12	0.618	207	4.97	0.622
0080	284	12.9%	162	0.58	0.109	164	0.61	0.126
0240	298	11.9%	295	0.64	3.479	297	0.64	3.485
0007	290	11.7%	283	1.53	0.150	287	1.51	0.148

Table 9: MegaDepth experiment. The table shows the **median** values of the rotation (in degrees) and translation errors. (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. Yellow represents the best result among the deep-based algorithms, and green among the classical algorithms.

Caama	N_c	Outliers%		Ours	.		RESf	M		Theia	ı		GLOM	AP
Scene	IV C	Outhers%	N_r	Rot	Trans	N_r	Rot	Trans	N_r	Rot	Trans	N_r	Rot	Trans
0238	522	44.6%	488	1.62	0.123	283	0.72	0.043	<u>506</u>	0.54	0.109	499	$\underline{0.22}$	0.043
0060	528	41.6%	518	$\underline{0.02}$	0.004	503	0.14	0.011	<u>525</u>	0.26	0.039	522	0.04	0.012
0197	870	40.7%	641	0.96	0.125	667	2.06	0.133	<u>855</u>	0.77	0.118	814	$\underline{0.13}$	0.016
0094	763	40.1%	663	0.26	0.019	537	0.38	0.015	<u>742</u>	0.21	0.033	717	$\underline{0.20}$	1.957
0265	571	38.8%	345	1.75	0.445	346	0.74	0.209	554	4.11	1.651	<u>558</u>	6.66	1.889
0083	635	31.3%	614	0.03	$\underline{0.002}$	596	0.15	0.009	<u>632</u>	0.15	0.013	614	0.04	0.007
0076	558	30.5%	543	$\underline{0.04}$	0.005	524	0.11	0.010	<u>549</u>	0.44	0.058	541	0.08	0.017
0185	368	30.0%	358	$\underline{0.04}$	$\underline{0.004}$	350	0.04	0.006	<u>365</u>	0.31	0.037	<u>365</u>	0.11	0.012
0048	512	24.2%	500	0.11	0.005	474	2.16	0.098	<u>507</u>	0.21	0.020	506	0.06	0.007
0024	356	23.0%	313	1.57	0.087	309	0.58	0.046	<u>355</u>	0.24	0.091	339	$\underline{0.07}$	$\underline{0.045}$
0223	214	17.0%	208	1.07	0.047	204	1.56	0.078	212	0.89	0.152	<u>214</u>	$\underline{0.41}$	$\underline{0.046}$
5016	28	16.9%	<u>28</u>	$\underline{0.04}$	0.003	<u>28</u>	0.10	0.005	<u>28</u>	0.07	0.019	<u>28</u>	$\underline{0.04}$	0.016
0046	440	14.6%	439	0.05	0.002	399	0.78	0.028	434	0.16	0.016	<u>440</u>	$\underline{0.02}$	$\underline{0.002}$
1001	285	43.9%	265	0.66	2.698	251	1.41	0.276	276	4.85	2.893	281	3.29	2.645
0231	296	42.2%	261	$\frac{0.00}{0.07}$	0.007	246	0.38	0.014	286	0.58	0.072	279	0.20	0.021
0411	299	29.9%	270	0.07	0.009	273	0.07	0.009	293	0.19	0.072	269	0.09	0.036
0377	295	27.5%	232	0.09	0.005	210	0.28	0.014	269	0.29	0.075	268	0.23	0.021
0102	299	25.8%	297	0.06	0.006	284	0.07	0.007	294	1.03	0.114	293	0.04	0.013
0147	298	24.6%	282	0.80	0.030	207	2.07	0.088	284	1.10	0.064	290	1.78	2.056
0148	287	24.6%	211	0.43	0.018	197	0.54	0.024	275	3.01	0.301	283	3.09	1.301
0446	298	22.1%	292	0.10	0.005	288	0.41	0.013	289	0.61	0.073	296	0.14	0.020
0022	297	21.2%	277	0.12	0.009	274	0.13	0.011	296	0.28	0.065	281	0.08	0.023
0327	298	21.0%	291	0.05	0.004	271	0.11	0.006	288	0.73	0.087	290	7.14	0.333
0015	284	20.6%	243	0.15	0.009	215	0.27	0.021	244	0.42	0.084	274	0.11	0.014
0455	298	19.8%	290	0.11	0.007	293	0.18	0.010	294	0.36	0.047	298	0.14	0.017
0496	297	19.2%	279	0.15	0.007	281	0.13	0.006	285	0.61	0.080	291	0.16	0.028
1589	299	17.4%	296	0.03	0.002	290	0.08	0.003	288	0.32	0.057	299	0.03	0.007
0012	299	16.3%	295	0.10	0.006	287	0.39	0.023	129	0.56	0.092	295	0.20	0.017
0019	299	15.4%	291	0.17	0.007	250	0.04	0.004	271	0.31	0.030	296	0.04	0.004
0063	293	14.5%	268	0.05	0.004	262	0.26	0.013	268	0.45	0.063	288	0.17	0.017
0130	285	14.4%	199	2.71	0.089	192	0.10	0.005	187	0.63	0.072	<u>281</u>	0.94	0.535
0080	284	12.9%	162	0.25	0.009	139	0.27	0.010	278	1.84	0.335	283	1.71	0.169
0240	298	11.9%	<u>295</u>	<u>0.10</u>	3.371	275	1.56	0.090	278	0.47	0.057	294	0.17	0.041
0007	290	11.7%	283	0.40	0.022	172	0.23	0.010	277	0.69	0.071	<u>290</u>	$\underline{0.06}$	<u>0.006</u>

Table 10: 1DSFM experiment. The table shows the **median** values of the rotation (in degrees), and translation errors. Winning results are marked in bold and underlined. Yellow represents the best result among the deep-based algorithms and green among the classical algorithms.

Scene	N_c	Outliers%		Ours	s		RESf	M		Theia		(GLOMA	AP
Scelle	IV c	Outliers%	N_r	Rot	Trans	N_r	Rot	Trans	N_r	Rot	Trans	N_r	Rot	Trans
Alamo	573	32.6%	509	$\underline{0.42}$	0.018	484	0.97	0.037	553	2.29	0.539	<u>557</u>	0.61	0.144
Ellis Island	227	25.1%	214	$\underline{0.16}$	0.033	214	0.32	0.036	213	3.85	0.712	<u>219</u>	0.46	0.087
Madrid Metropolis	333	39.4%	295	0.27	0.016	244	4.42	0.193	-	-	-	<u>320</u>	0.53	0.096
Montreal Notre Dame	448	31.7%	425	$\underline{0.16}$	$\underline{0.020}$	346	1.00	0.056	422	2.63	0.808	<u>444</u>	0.40	0.158
NYC Library	330	33.6%	285	0.58	0.038	224	1.48	0.074	<u>534</u>	1.65	0.360	323	$\underline{0.46}$	0.075
Notre Dame	549	35.6%	519	$\underline{0.29}$	$\underline{0.012}$	517	0.55	0.025	314	1.54	0.133	$\underline{543}$	1.15	0.130
Piazza del Popolo	336	33.1%	315	2.11	0.120	249	0.80	$\underline{0.034}$	325	1.15	0.342	<u>331</u>	$\underline{0.28}$	0.084
Tower of London	467	27.0%	454	$\underline{0.23}$	0.011	94	0.48	0.012	448	3.23	0.527	<u>466</u>	0.42	0.071
Vienna Cathedral	824	31.4%	753	11.78	0.527	479	0.48	0.016	772	9.32	0.838	822	0.61	0.206
Yorkminster	432	29.0%	403	0.62	$\underline{0.022}$	331	4.67	0.299	390	4.26	0.948	<u>418</u>	$\underline{0.60}$	0.069