QUACAMFM: QUATERNION-CONSTRAINED FLOW MATCHING FOR CAMERA POSE ESTIMATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Camera pose estimation from multi-view images remains a challenge in computer vision. Traditional methods often address this problem using Structure-from-Motion (SfM) with bundle adjustment. However, camera poses estimated from sparse views are inherently ambiguous due to insufficient geometric constraints. Recent work leverages probabilistic models, such as diffusion models, to generate multiple camera pose hypotheses and therefore capture this uncertainty better. Most of these methods represent camera rotations using unit quaternions, but treat them as unconstrained 4D vectors during the generative processes, thereby ignoring the unit-norm constraint of quaternions. Unconstrained quaternions create non-smooth and suboptimal generation trajectories. To this end, we propose *QuacamFM*, a quaternion-constrained flow matching framework for camera pose estimation that preserves unit quaternion representations throughout the entire flow trajectory. We design the optimal transport of the quaternion flows using smooth spherical linear interpolation. Experiments on CO3Dv2 demonstrate our method's advantage in camera pose accuracy over diffusion-based methods and classical SfM approaches. We further show that our quaternion-constrained formulation outperforms the naive application of standard flow matching to 4D quaternion vectors on sparse-view camera pose estimation. Finally, it is observed that QuacamFM generalizes well across datasets and in-the-wild examples. The code is released upon acceptance.

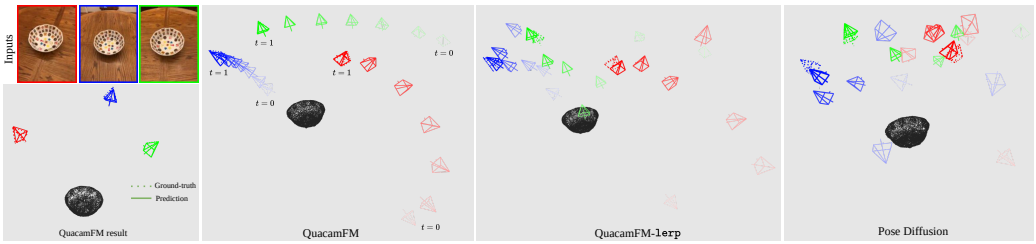

Figure 1: **Camera generation trajectory** of **QuacamFM** vs. QuacamFM-`lerp` and PoseDiffusion during inference, from starting step $t = 0$ to ending step $t = 1$. Lighter shades indicate earlier flow steps, and darker shades indicate later steps. QuacamFM produces smoothest trajectories, yielding better camera pose estimation accuracy.

1 INTRODUCTION

Recovering camera poses, *i.e.* translations and rotations from an unordered set of multi-view images is a fundamental problem in computer vision (Agarwal et al., 2011). Accurate camera poses establish the essential geometric relationships in many downstream applications, including 3D reconstruction, visual localization, and robotic perception (Mur-Artal & Tardos, 2017; Wang et al., 2024a; Duisterhof et al., 2025). Traditionally, camera poses are estimated via Structure-from-Motion (SfM) (Schonberger & Frahm, 2016), which recovers camera geometry by exploiting geometric constraints on image correspondences and a bundle adjustment process in a global coordinate frame. Recent advances in keypoint detection, feature matching, and correspondence estimation (Sarlin et al.,

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

2020; Edstedt et al., 2024b; Leroy et al., 2024) significantly strengthen SfM pipelines, especially under illumination changes.

Despite these advances, SfM remains challenging in sparse-view settings, where limited image overlaps prevent the establishment of sufficiently reliable correspondences between image pairs. Regression-based approaches overcome this by directly predicting camera poses or intermediate geometric representations from image observations without explicitly establishing pairwise correspondences, and have shown substantial improvements in sparse-view camera estimation (Sinha et al., 2023; Zhang et al., 2022; Wang et al., 2024b). However, deterministically regressing camera poses from sparse views is inherently ambiguous due to insufficient geometric constraints. Probabilistic approaches have recently emerged as a promising alternative, particularly in challenging scenarios involving symmetric objects or image pairs with limited overlap (Wang et al., 2023; Zhang et al., 2024; Zhao et al., 2025), to better capture the uncertainty by generating multiple plausible camera pose hypotheses. Recently, PoseDiffusion (Wang et al., 2023) leverages the diffusion framework (Ho et al., 2020) to jointly model camera rotation and translation. Similar to Sinha et al. (2023); Kendall et al. (2015); Shavit et al. (2021), PoseDiffusion uses 4D quaternions to represent camera rotations, but treating them as unconstrained 4D vectors and additively inject/remove noise during their generation. The additive generation process breaks the unit-length constrain of quaternions, causing them to no longer represent pure 3D rotations, leading to decreases in camera rotation estimation accuracy.

In this paper, we propose *Quaternion Camera Flow Matching* (QuacamFM), a novel probabilistic approach for sparse multi-view camera pose estimation, using an operation based on the Hamilton product to explicitly ensure the 3D-rotation validity of camera quaternion poses. We learn the optimal-transport (OT) mapping from a source distribution, *e.g*. the Gaussian distribution, to the target ground-truth camera quaternion pose distribution, and the velocity is modeled via a quaternion-valued differential equation (QDE). When solving the QDE, unlike additions, the Hamilton product preserves unit-quaternions throughout the mapping trajectory by operating on the unit sphere manifold $\mathcal{S}^3$. We define the desired optimal transport of a randomly sampled quaternion to its ground-truth target as the *spherical linear interpolation* (`slerp`). To further highlight our proposed method, we also consider a vanilla formulation, QuacamFM-`lerp`, that directly applies standard flow matching, with linear interpolation as the desired OT mapping, and the quaternions are normalized to unit norm at every integration step to yield valid rotations. A demonstration of our proposed method showing smooth camera OT trajectory during inference can be seen in Fig. 1. We train and evaluate our model on the CO3Dv2 dataset and show clear advantages of using `slerp` as OT trajectory over the vanilla `lerp` in camera rotation recovery. We further assess its generalization on the held-out CO3Dv2 object categories and the scene-centric RealEstate10K dataset. We also demonstrate the applicability of our method to in-the-wild self-captured data. To summarize, our contributions are listed as follows:

- We propose QuacamFM, a novel quaternion-constrained flow-matching framework for sparse multi-view camera pose estimation that preserves valid camera rotations throughout the flow trajectory.
- We empirically show that enforcing the quaternion constraint in flow matching significantly improves camera rotation accuracy compared with naively treating quaternions as unconstrained 4D vectors in normalizing flows.
- Our method demonstrates significant improvements over previous methods, including classic SfM and diffusion-based approaches on CO3Dv2, while also generalizing well to unseen scenes from RealEstate10K and in-the-wild captures.

2 Related work

Structure from Motion. SfM is a long-standing problem in computer vision (Hartley et al., 2003; Özyeşil et al., 2017), aiming to estimate camera parameters from image sequences through global (Cui & Tan, 2015; Moulon et al., 2013; Wilson & Snavely, 2014) or incremental (Agarwal et al., 2011; Schonberger & Frahm, 2016; Snavely et al., 2006; Wu, 2013) pipelines. Its principal components, including keypoint detection, feature matching, and reconstruction, have been substantially improved by learning-based techniques (Edstedt et al., 2024a;b; Nordström et al., 2026; Sarlin et al., 2019). Improvements in these individual components have boosted the overall performance of the SfM framework. Feed-forward SfM methods have also recently emerged (Wang

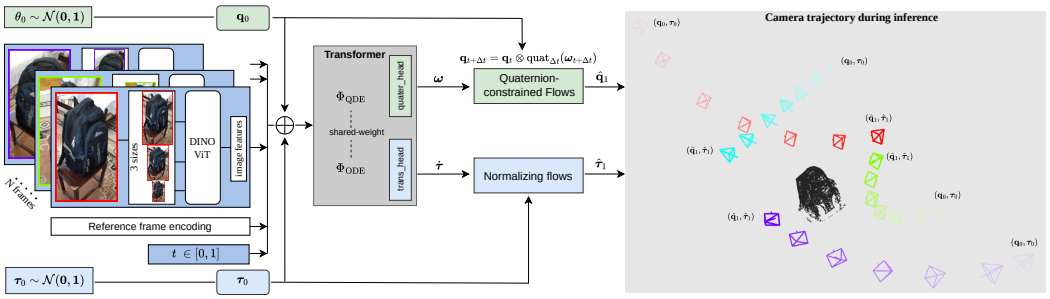

Figure 2: **QuacamFM overview**. We present a flow-matching-based method to predict camera poses given an image sequence. The camera rotation is estimated via quaternion-constrained flows, where the optimal transport path is constructed using `slerp`, preserving the unit-quaternion constraint. The quaternion trajectory is integrated using the quaternion differential equation, while the translation uses a standard normalizing flow. The rightmost picture shows the camera trajectory during inference, lighter shades indicate earlier flow steps, and darker shades indicate later steps.

et al., 2024a; 2025; Elflein et al., 2025; Zhang et al., 2025a; Duisterhof et al., 2025; Wang et al., 2024b) to directly learn 3D geometry rather than through iterative optimization. DUSt3R (Wang et al., 2024b) and MASt3R (Leroy et al., 2024) leverage large-scale training data to regress dense 2D-3D corresponding pointmaps, from which camera poses and various other geometric quantities can be recovered. However, camera pose estimates in sparse-view unordered image settings still remain challenging due to limited pair-wise geometric constraints. Regression-based approaches are proposed (Kendall et al., 2015; Sinha et al., 2023; Brahmbhatt et al., 2018) to learn the camera poses end-to-end. Direct regression of camera rotation and translation has been explored by Wang et al. (2023), showing improved performance over the classical COLMAP pipeline when fewer than ten input views are available. SparsePose (Sinha et al., 2023) further improves upon naive pose regression by using the regressed camera poses as an initial estimate and subsequently refining them through a learnable iterative module that exploits 3D projective-consistent local features.

Probabilistic model for Camera pose estimation. Probabilistic models have emerged as a promising approach for handling uncertainty in sparse-view settings. RelPose (Zhang et al., 2022) adopts an energy-based formulation to model distributions over relative camera rotations, enabling the estimation of a consistent set of camera rotations. RelPose++ (Lin et al., 2024) builds upon this formulation with a more modern architecture and extends it to jointly estimate camera translations. DiffusionSfM (Zhao et al., 2025) applies diffusion to optical rays and their endpoints to recover camera geometry. Camera-as-Rays (Zhang et al., 2024) formulates camera as distribution of optical rays in Plücker coordinates. PoseDiffusion (Wang et al., 2023) learns to denoise camera rotations and translations directly from random noise, complemented by geometric guidance during inference. However, PoseDiffusion formulates quaternions of the camera rotations as independent unconstrained 4D vectors. We propose a novel method using flow-matching framework to explicitly preserve the quaternion constraint for sparse unordered multi-view camera pose estimation.

3 METHOD

From an image sequence $I = (I^i)_{i=1}^N$ of a scene, where $N \in \mathbb{N}$ is the number of input images and each image $I^i \in \mathbb{R}^{3\times H\times W}$, we estimate the camera extrinsics, *i.e.* camera translations and rotations. Each corresponding camera pose is represented as $(\mathbf{q}^i, \boldsymbol{\tau}^i)$, where a unit quaternion $\mathbf{q}^i \in \mathbb{R}^4$ denotes the camera rotation and $\boldsymbol{\tau}^i \in \mathbb{R}^3$ denotes the camera translation. The camera extrinsic parameterization in the canonical coordinate system is detailed in Section 3.4.

Fig. 2 presents the overview of our quaternion-constrained flow matching framework. For camera rotation, we learn the optimal transport (OT) angular velocity, *i.e.* quaternion flows, to map randomly sampled camera rotations to their corresponding ground-truths, under the unit-sphere constraint of quaternions. For camera translation, we learn the OT velocity in Euclidean space, *i.e.* normalizing flows, for the translational mapping. Details are described in the following sections.

3.1 Quaternion Differential Equation

A quaternion $\mathbf{q} \in \mathbb{R}^4$ is described by a 4D vector $(\mathrm{w}, v_1, v_2, v_3)$, expressed as:

$$\mathbf{q} = \mathrm{w} + v_1 i + v_2 j + v_3 k = (\mathrm{w}, \mathbf{v}),$$

where i, j, k are imaginary units satisfying $i^2 = j^2 = k^2 = -1$. To correctly represent a 3D rotation, this 4D vector must be a unit quaternion, *i.e.* normalized to unit length, $\|\mathbf{q}\| = 1$. We refer to all unit quaternions as simply quaternions for convenience in this paper. The Hamilton product $\otimes$ combines two quaternions, $\mathbf{q}_1 = (\mathrm{w}_1, \mathbf{v}_1)$ and $\mathbf{q}_2 = (\mathrm{w}_2, \mathbf{v}_2)$, into a new quaternion as $\mathbf{q}_1 \otimes \mathbf{q}_2 = \big(\mathrm{w}_1\mathrm{w}_2 - \mathbf{v}_1^\top \mathbf{v}_2, \mathrm{w}_2\mathbf{v}_1 + \mathrm{w}_1\mathbf{v}_2 + \mathbf{v}_1 \times \mathbf{v}_2\big)$. The Hamilton product is non-commutative, *i.e.* $\mathbf{q}_1 \otimes \mathbf{q}_2 \neq \mathbf{q}_2 \otimes \mathbf{q}_1$. Following prior work (Le et al., 2026; Gołąbek et al., 2022; Kuipers, 1999), the quaternion flow is described by a quaternion-value ordinary differential equation (QDE) as:

$$\dot{\mathbf{q}} = \frac{1}{2} \begin{bmatrix} -[\boldsymbol{\omega}]_\times & \boldsymbol{\omega} \\ -\boldsymbol{\omega}^\top & 0 \end{bmatrix} \mathbf{q}, \quad [\boldsymbol{\omega}]_\times = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix}, \tag{1}$$

where $\boldsymbol{\omega} \in \mathbb{R}^3$ is the angular velocity vector, whose direction and magnitude represent the rotation axis and angular speed, respectively.

Solution to the QDE. In conventional flow matching, the state is evolved by integrating the predicted velocity field using additive ordinary differential equation (ODE) updates. However, for quaternion flows, the quaternion representation of a camera rotation $\mathbf{q}_t$ is constrained to the manifold $\mathcal{S}^3$, and a vanilla additive ODE update does not preserve the unit-norm constraint. Therefore, to solve the QDE numerically, the Hamilton product is required to integrate the predicted velocity field while ensuring the quaternion solution remains on $\mathcal{S}^3$. We implement multiplicative quaternion integration via the Hamilton product as:

$$\mathbf{q}_{t+\Delta t} = \mathbf{q}_t \otimes \mathrm{quat}_{\Delta t}(\boldsymbol{\omega}_{t+\Delta t}) = \mathbf{q}_t \otimes \left(\cos\left(\frac{\theta}{2}\right), \sin\left(\frac{\theta}{2}\right) \frac{\boldsymbol{\omega}_t}{\|\boldsymbol{\omega}_t\|} \right), \tag{2}$$

where $\theta = \|\boldsymbol{\omega}_t\| \Delta t$ is the scalar magnitude of the rotation scaled by a time step Δt, $\boldsymbol{\omega}_t$ is normalized to a unit vector, and $\mathrm{quat}_{\Delta t}(\cdot)$ is the quaternionized angular velocity. This integration scheme preserves the unit-norm constraint of quaternions on $\mathcal{S}^3$, resulting in smooth and continuous generation trajectories for camera rotations. We use this integration in our QuacamFM, and Section 4 shows superior results in camera rotation accuracy compared to naively using normalizing flows, see Fig. 5.

3.2 Flow Matching for Camera

From the input images I, QuacamFM learns to jointly estimate the OT rotation and translation velocities to recover the ground-truth camera rotations $\mathbf{q}_1$ and camera translations $\boldsymbol{\tau}_1$. We use the extracted visual features $\psi(I)$ from DINO-ViT (Caron et al., 2021) as the generative flow condition, and the details of this extraction, including how we handle the coordinate ambiguity, are provided in Section 3.4. The estimation of camera rotation and translation are described in the following sections, see also Fig. 2 for an overview.

Quaternion Flow Matching. Given the visual features of image inputs $\psi(I)$, we seek to learn the optimal transport from random to the target camera quaternion rotation. Consider the mapping over the time interval $t \in [0, 1]$ from an initial random quaternion sample $\mathbf{q}_0$ to the ground-truth target quaternion $\mathbf{q}_1$, the diffeomorphic flow from $\mathbf{q}_0$ to $\mathbf{q}_1$ is defined via the QDE in Eq. (1), and the angular velocity is estimated as:

$$\boldsymbol{\omega}_t = \Phi_{\mathrm{QDE}}(\mathbf{q}_t, t, \psi(I)), \tag{3}$$

where Φ_{QDE} is the parameterized model, here chosen as the Transformer architecture (Vaswani et al., 2017), and $t \in [0, 1]$ is the timestamp. Common flow matching pipelines (Lipman et al., 2023; Tong et al., 2023; Zhang et al., 2025b) often adopt a linear interpolation between source and target samples as the OT conditional probability path, due to its simplicity and stable training behavior. As explained in Section 3.1, quaternions must stay on the unit sphere $\mathcal{S}^3$ to correctly represent rotations,

we identify the OT path as a *spherical linear interpolation* (`slerp`) when learning camera rotations. The interpolated value $\mathbf{q}_t$, using `slerp` between $\mathbf{q}_0$ and $\mathbf{q}_1$, at time t is defined as:

$$\mathbf{q}_t = \texttt{slerp}(\mathbf{q}_0, \mathbf{q}_1, t) = \mathbf{q}_0 \otimes (\mathbf{q}_0^{-1} \otimes \mathbf{q}_1)^t \ . \tag{4}$$

To instantiate a quaternion $\mathbf{q}_0$ for a valid 3D rotation, we sample a random axis-angle rotation $\boldsymbol{\theta}_0 \in \mathbb{R}^3$ from the Gaussian distribution, $\boldsymbol{\theta}_0 \sim \mathcal{N}(0, \mathbf{I}_3)$, and then convert it to quaternion $\mathbf{q}_0$. Following that, the OT quaternion derivative $\dot{\mathbf{q}}^*$ is calculated as:

$$\dot{\mathbf{q}}^* = \log(\mathbf{q}_0^{-1} \otimes \mathbf{q}_1) \ , \tag{5}$$

where $\boldsymbol{\omega}^*$ is the corresponding angular velocity, derived by taking only the vector part of $\dot{\mathbf{q}}^*$ (Sola, 2017; Dunn & Parbery, 2011). Note that `slerp` is a linear interpolation with a constant derivative, thus $\boldsymbol{\omega}^*$ is independent of time. Inspired by the work from Yue et al. (2025), we learn to predict the angular velocity $\boldsymbol{\omega}_t = \Phi_{\text{QDE}}(\mathbf{q}_t, t, \psi(I))$ via a regression objective:

$$\mathcal{L}_\omega = \|\boldsymbol{\omega}_t - \boldsymbol{\omega}^*\|_1 \ , \ \forall t \in [0, 1] \ . \tag{6}$$

Vanilla Quaternion Transport. To highlight the importance of quaternion OT flow matching, we also provide a variant (QuacamFM-`lerp`) using the vanilla linear transport `lerp`, which treats the camera quaternion rotation as an un-constrained 4D vector. We keep other components unchanged, but the interpolated quaternion and the ground-truth quaternion derivative in Eq. (4) and Eq. (5) are replaced with:

$$\begin{aligned} \mathbf{q}_t &= \texttt{lerp}(\mathbf{q}_0, \mathbf{q}_1, t) = (1-t)\mathbf{q}_0 + t\,\mathbf{q}_1 \ , \\ \Rightarrow \dot{\mathbf{q}}^*_{\text{lerp}} &= \mathbf{q}_1 - \mathbf{q}_0 \ . \end{aligned} \tag{7}$$

The angular velocity $\boldsymbol{\omega}^*$ calculation and learning objective remain the same as when using `slerp`.

Translation Flow Matching. Following the common flow matching formulation for unconstrained variables, we model the translation using a linear interpolation (`lerp`) conditional path and an additive ODE solver. The initial sample $\boldsymbol{\tau}_0 \in \mathbb{R}^3$ is sampled from the Gaussian distribution, $\boldsymbol{\tau}_0 \sim \mathcal{N}(0, \mathbf{I}_3)$. The transport trajectory and the OT translation velocity are then given by:

$$\begin{aligned} \boldsymbol{\tau}_t &= \texttt{lerp}(\boldsymbol{\tau}_0, \boldsymbol{\tau}_1, t) = (1-t)\,\boldsymbol{\tau}_0 + t\,\boldsymbol{\tau}_1 \ , \\ \Rightarrow \dot{\boldsymbol{\tau}}^* &= \boldsymbol{\tau}_1 - \boldsymbol{\tau}_0 \ . \end{aligned} \tag{8}$$

The transport velocity estimator of the translation vector is a regression model $\dot{\boldsymbol{\tau}}_t = \Phi_{\text{ODE}}(\boldsymbol{\tau}_t, t, \psi(I))$, supervised by the learning objective:

$$\mathcal{L}_{\dot{\boldsymbol{\tau}}} = \|\dot{\boldsymbol{\tau}}_t - \dot{\boldsymbol{\tau}}^*\|_1 \ , \ \forall t \in [0, 1] \ . \tag{9}$$

3.3 Solving the camera extrinsic.

Once the transport velocities $\boldsymbol{\omega}$ and $\dot{\boldsymbol{\tau}}$ are estimated, the camera rotation and translation can be solved for numerically. In particular, the quaternion vector and translation vector are integrated separately, corresponding to the QDE in Eq. (1) and an ODE. Using the Euler integration scheme, we solve the camera rotation via the numerical integration with predicted $\boldsymbol{\omega}$, expressed as:

$$\mathbf{q}_{t+\Delta t} = \mathbf{q}_t \otimes \text{quat}_{\Delta t}(\boldsymbol{\omega}) \ , \tag{10}$$

where $\otimes$ denotes the Hamilton product, $\psi(I)$ is the input image feature, and $\text{quat}_{\Delta t}(\boldsymbol{\omega})$ is the quaternionized angular velocity following Eq. (2). The camera translation is obtained by integrating the ODE using the Euler solver:

$$\boldsymbol{\tau}_{t+\Delta t} = \boldsymbol{\tau}_t + \dot{\boldsymbol{\tau}} \Delta t. \tag{11}$$

At the end of these integration procedures, we acquire the estimated quaternion representation $\hat{\mathbf{q}}_1$ of the camera rotation and the estimated camera translation $\hat{\boldsymbol{\tau}}_1$. Note that, in the variant QuacamFM-`lerp`, which treats quaternion as an un-constrained 4D vector and uses vanilla `lerp` as the optimal transport, the integration is updated identically as Eq. (11) but replacing $\boldsymbol{\tau}$ with $\mathbf{q}$.

The total learning objective is the aggregation of all the transport velocity loss in Eq. (6), Eq. (9), and the reconstructed camera extrinsic loss over camera poses. In which, the camera extrinsic losses, $\mathcal{L}_{\mathbf{q}}$ and $\mathcal{L}_{\boldsymbol{\tau}}$, are computed against the ground-truth poses, supervising the estimated rotation and translation, respectively. With the hyperparameters λ_ω, $\lambda_{\dot{\boldsymbol{\tau}}}$, $\lambda_{\mathbf{q}}$, $\lambda_{\boldsymbol{\tau}}$ balance the contributions of the loss terms, the total loss is computed as:

$$\mathcal{L} = \lambda_\omega \mathcal{L}_\omega + \lambda_{\dot{\boldsymbol{\tau}}} \mathcal{L}_{\dot{\boldsymbol{\tau}}} + \lambda_{\mathbf{q}} \mathcal{L}_{\mathbf{q}} + \lambda_{\boldsymbol{\tau}} \mathcal{L}_{\boldsymbol{\tau}} \ . \tag{12}$$

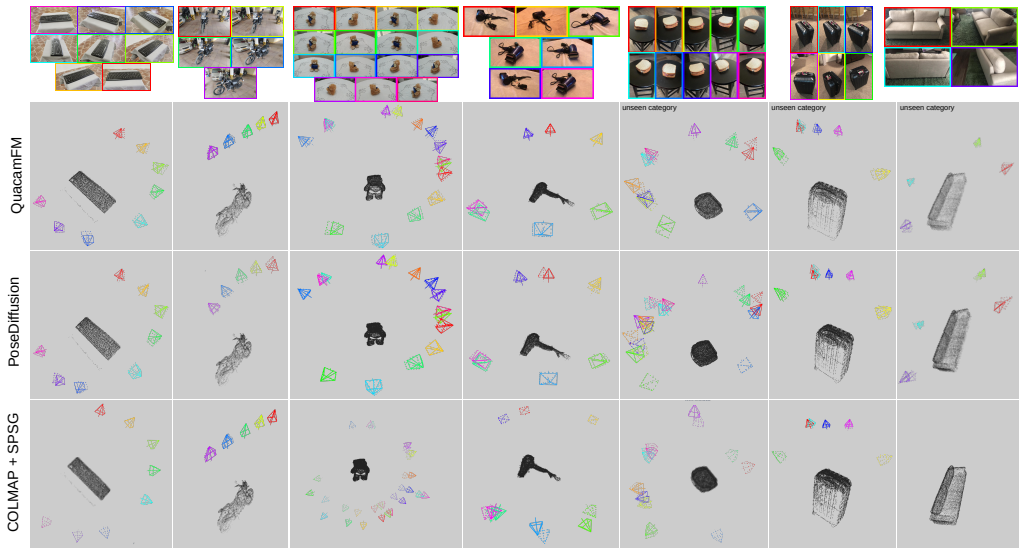

Figure 3: **Qualitative results on CO3Dv2**. Camera pose estimations from input images (top row). Our QuacamFM and baseline results are shown in below rows. Camera frustum colors correspond to input-image borders. Solid frustums are camera estimations and dash frustums are camera ground-truths. The first four examples are from seen categories, and last three from held out (unseen) categories. Missing cameras indicate non-convergence.

3.4 Implementation Details

Coordinate Ambiguity. Inspired by the work of Wang et al. (2023), to avoid overfitting to a certain scene coordinate, we randomly select N input images, and use one of them as the reference. This means the reference camera is transformed to the origin, resulting in an identity rotation. We transform the inputs to canonical form with respect to the reference-camera transformation. This preserves the relative poses of the input cameras with respect to the randomly selected reference frame. Since camera translation is only determined up to an arbitrary scale, we normalize the translations by dividing them by the median of the norms of the transformed translation vectors, same as the variant Quacam-lerp. To further prevent over-fitting to a certain number of input frames, we train with dynamic input frames, *i.e*. randomly chose between 3-30 frames at each training step.

Experimental Setting. We use the pre-trained DINO ViT (Caron et al., 2021) to extract input image features $\psi(I)$. Following the setup of Wang et al. (2023), we crop the input images around the object center and resize them to 224×224. We then extract visual features at multiple scales of 1, $1/2$, and $1/3$ and average the resulting features. The weights of DINO are enabled to be optimized during training. To allow the model to identify the selected reference frame, we concatenate its extracted visual feature with a one-hot indicator set to 1 along the last channel. We adopt the Transformer architecture from Vaswani et al. (2017) for the shared-weight transport velocity estimator of Φ_{ODE} and Φ_{QDE}. The estimator has 8 encoder layers, with 8 attention heads, and the hidden dimension of the feedforward network is 1024. We use the multi-layer perceptron (MLP) for the quaternion head and translation head in Fig. 2. The hyper-parameters are set to $\lambda_\omega = 1.0$, $\lambda_{\boldsymbol{\tau}} = 1.0$, $\lambda_{\mathbf{q}} = 0.1$, $\lambda_{\boldsymbol{\tau}} = 2.0$. To accelerate the training process, we repeat input batches, allowing the model to see a batch at multiple random timesteps at once.

4 Experiments

Datasets. We use two datasets with different characteristics. 1) The object-centric CO3Dv2 (Reizenstein et al., 2021), containing about 37k turntable-style videos from 51 MS-COCO categories Lin et al. (2014), with camera poses estimated by COLMAP (Schonberger & Frahm, 2016) from 200 frames. Following Zhang et al. (2022); Wang et al. (2023), we train on 41 categories only, and hold-out the remaining 10 categories for evaluating generalization to unseen object categories.

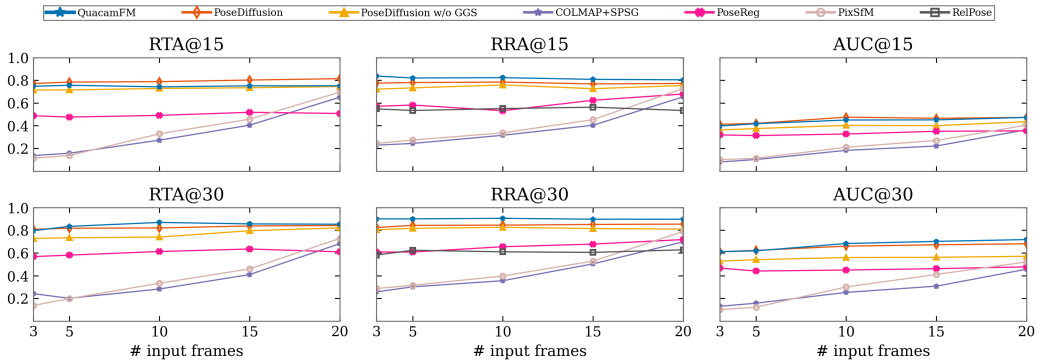

Figure 4: **Camera pose accuracy on CO3Dv2**. Quantitative metrics (y-axis, higher-better) at different thresholds and numbers of input images (x-axis). RelPose does not predict camera translation and hence is omitted in the respective metrics.

Table 1: **Results on CO3Dv2**. Quantitative evaluation of camera pose estimation with 10-frame input. **Bold** denotes the best and underline the second best. The inference time is measured in seconds (s). †Methods that require correspondences or post optimization.

Method	RTA@30↑	RRA@30↑	AUC@30↑	RTA@15↑	RRA@15↑	AUC@15↑	s
PixSfM	40.4	38.6	30.1	32.9	33.7	28.3	20.2
COLMAP + SPSG	28.2	44.4	25.1	27.3	31.6	21.1	3.26
PoseReg	61.3	65.5	45.5	49.1	53.2	35.7	0.03
RelPose†	-	72.7	-	-	57.2	-	29.5
RelPose++†	84.3	88.7	66.5	74.2	**84.5**	45.1	4.49
PoseDiffusion†	79.9	86.7	66.5	**79.8**	80.5	**47.5**	8.83
PoseDiffusion w/o GGS	72.8	82.8	56.0	67.8	75.2	43.3	0.39
QuacamFM-`lerp`	77.2	75.8	51.3	60.4	67.8	41.3	0.46
QuacamFM	**86.7**	**90.5**	**67.3**	74.2	82.0	45.1	0.46

2) The scene-centric RealEstate10K (Zhou et al., 2018), with around 80k YouTube videos of indoor and outdoor scenes; its camera trajectories are estimated by ORB-SLAM2 (Mur-Artal et al., 2015) and refined with bundle adjustment. Following Wiles et al. (2020); Wang et al. (2023), we evaluate on randomly sampled 1.8k videos from the original 7k-video test set, without further training.

Metrics. To evaluate the performance, we use Relative Rotation Accuracy (RRA) and Relative Translation Accuracy (RTA). The RRA is used to compare the relative rotation $R_i R_j^T$ from i-th camera to j-the camera to the ground truth $R_i^* R_j^{*T}$. The RTA defines the angle between the predicted translation and the ground-truth $\boldsymbol{\tau}_{ij}/\boldsymbol{\tau}_{ij}^*$ pointing from camera i to camera j as $\text{RTA}(\boldsymbol{\tau}_{ij}, \boldsymbol{\tau}_{ij}{}^*) = \texttt{arccos}(\boldsymbol{\tau}_{ij}{}^T \boldsymbol{\tau}_{ij}{}^* / \|\boldsymbol{\tau}_{ij}\| \|\boldsymbol{\tau}_{ij}{}^*\|)$. Both metrics are invariant to the choice of global coordinate frame. Following Wang et al. (2023); Jin et al. (2021), we also report the Area Under the Curve (AUC) metric, which summarizes pose accuracy over a range of angular thresholds for both RRA and RTA. AUC defines the accuracy at a threshold δ as $\texttt{min}(\text{RTA@}\delta, \text{RRA@}\delta)$.

Baselines. We compare against COLMAP+SPSG (Schonberger & Frahm, 2016), using robust SuperPoint (DeTone et al., 2018) keypoint detector and SuperGlue (Sarlin et al., 2020) matcher. PixSfM (Lindenberger et al., 2021) improves reconstruction accuracy through direct alignment of deep features. For regression-based estimation, we include PoseReg, following the regression baseline introduced in Wang et al. (2023). We also evaluate against RelPose (Zhang et al., 2022), which models a distribution over camera rotations, and RelPose++ (Lin et al., 2024), which extends RelPose to jointly estimate camera translations. Importantly, we compare with PoseDiffusion (Wang et al., 2023), a state-of-the-art probabilistic method for sparse-view camera pose estimation that incorporates geometric guidance on correspondences during the diffusion sampling process. For a fair comparison, we also report PoseDiffusion without geometric guidance (PoseDiffusion w/o GGS),

Table 2: Results on the **held-out** CO3Dv2 categories with 10 input frames.

Method	RTA@15	RRA@15	AUC@30
COLMAP + SPSG	43.6	35.8	30.3
PixSfM	38.3	38.9	34.2
PoseReg	42.2	45.5	36.4
PoseDiffusion†	**55.8**	62.8	50.8
PoseDiffusion w/o GGS	48.8	56.1	40.1
QuacamFM-`lerp`	54.6	60.3	45.5
QuacamFM	**55.8**	**64.9**	**51.1**

Table 3: Results on the RealEstate10K with 10 input frames.

Method	RTA@15	RRA@15	AUC@30
COLMAP + SPSG	60.2	72.8	45.2
PixSfM	64.2	**78.3**	**49.4**
PoseReg	35.4	55.6	35.2
PoseDiffusion†	**67.2**	76.8	48.0
PoseDiffusion w/o GGS	38.7	64.3	28.7
QuacamFM-`lerp`	50.1	70.6	40.3
QuacamFM	50.6	**78.3**	42.7

which relies solely on the learned diffusion model and does not require explicit correspondence. Methods that require post optimization or access to correspondences are marked with †.

We train our method only on CO3Dv2 and evaluate it on the test set, unseen (held-out) CO3Dv2 categories, and RealEstate10K. Results are reported in Section 4.1. To further highlight the contribution of our quaternion-constrained flow formulation, we compare QuacamFM against a vanilla flow-matching baseline, QuacamFM-`lerp`, in Section 4.2.

4.1 Pose Estimation

Result on CO3Dv2. We compare QuacamFM with baseline methods on CO3Dv2 using varying numbers of input images. Fig. 4 presents quantitative results for 3 to 20 input views, while Table 1 reports camera pose accuracy with 10 input images, and the inference time is measured on a `NVIDIA-A100` GPU. Our method substantially outperforms the SfM-based approaches, COLMAP+SPSG and PixSfM, as well as the direct regression baseline, PoseReg. It further achieves the best performance across all metrics at the threshold of @30 in both sparse- and dense-view settings, while remaining competitive at the stricter threshold @15. PoseDiffusion achieves higher translation accuracy on RTA@15 due to its geometric guidance (GGS) over correspondences, but costs additional computation from iterative optimization over the last 10 denoising steps as in the standard setting (Wang et al., 2023), *e.g.* significant increase from 0.39s to 8.83s. RelPose++ achieves relatively higher rotation accuracy at RRA@15, but also requires an additional optimization stage to enforce mutual geometric consistency among the predicted camera rotations. We use a common choice of 50 optimization iterations (Zhang et al., 2024), taking approximately 4.49s. A fairer comparison is PoseDiffusion w/o GGS, which has a similar model capacity and does not have access to correspondences, consuming a runtime of 0.4s, roughly similar to our method. Our method significantly outperforms this baseline, as shown in Fig. 4. Furthermore, Table 1 clearly shows that QuacamFM achieves substantially better camera accuracy than its variant, QuacamFM-`lerp`. This demonstrates that intuitively applying a standard flow-matching pipeline to camera pose estimation is not effective, while explicitly imposing the quaternion constraint leads to significantly improved performance. Further detailed analysis of `slerp` versus `lerp` is provided in Section 4.2.

Generalization to unseen CO3Dv2 and RealEstate10K. We assess the generalization ability of our method on held out CO3Dv2 object categories and the RealEstate10K dataset. RealEstate10K is scene-centric and follows a data distribution that is unseen during training. It contains free-form camera motion, in contrast to the object-centric orbital camera motion in CO3Dv2. Following the evaluation protocol in Wang et al. (2023), we report the results in Table 2 and 3. Our method remains on par with PoseDiffusion, which has access to image correspondences, and outperforms the other baselines. Qualitative examples on RealEstate10K dataset can be seen in the Appendix.

4.2 Importance of Quaternion-constrained flows

To validate the importance of quaternion-constrained flow, we provide the variant QuacamFM-`lerp` using the same architecture and training hyper-parameters, but ablating the rotation optimal transport path. We mainly evaluate the performance on rotation accuracy, using stricter thresholds (@5, @15, @30) with a wider range of input views to assess performance consistency. We also compare against other probabilistic methods, *i.e.* PoseDiffusion and PoseDiffusion w/o GGS, in Fig. 5.

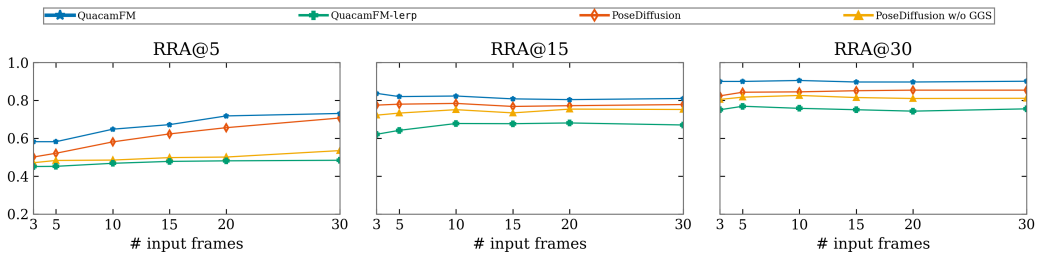

Figure 5: **Optimal transport ablation**. Comparison of rotation accuracy between quaternion-constrained flows (QuacamFM), vanilla normalizing flows (QuacamFM-`lerp`), and diffusion-based methods (PoseDiffusion and PoseDiffusion w/o GGS). Experiments conducted on CO3Dv2 test set.

Fig. 5 demonstrates that naively treating quaternions as unconstrained 4D vectors (QuacamFM-`lerp`) in the flow-matching framework does not yield any improvements over the straightforward diffusion-based approach, *i.e*. PoseDiffusion w/o GGS. It remains notably behind PoseDiffusion, which has access to image pairwise correspondences during denoising process. QuacamFM, imposing the unit-constraints and using QDE, shows significant rotation accuracy improvement consistently under multiple thresholds and input image ranges, without access to correspondences. The camera evolvement animation during integration of QuacamFM, QuacamFM-`lerp` and during denoising of PoseDiffusion is illustrated in Appendix materials.

4.3 Demonstration on In-the-wild inputs

Finally, we further demonstrate generalization beyond the training data distribution by presenting results on in-the-wild self-captured sequences. We show results with varying numbers of input images and demonstrate that our method can recover plausible camera poses that are visually consistent with the input views, as shown in Fig. 6.

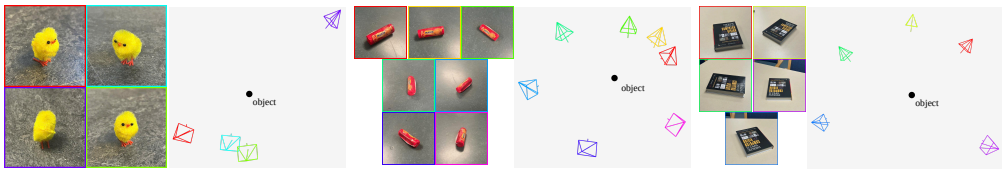

Figure 6: **Camera pose estimations on in-the-wild examples**. QuacamFM generalizes well to images outside of the distribution of CO3Dv2 training categories, and can recover plausible camera poses even on self-captures that contain varying distances from the camera and in-plane rotations.

5 Conclusion

This paper presents QuacamFM, a quaternion-constrained flow-matching framework for sparse multi-view camera pose estimation. QuacamFM uses `slerp` to design the optimal transport path, thus preserving valid camera rotations throughout the flow trajectory. By using integration of the quaternion differential equation, it enables smooth and geometrically valid rotational flows. Our experiments show QuacamFM outperforms diffusion-based approaches and traditional SfM methods, such as COLMAP. We further demonstrate that explicitly enforcing the quaternion constraint improves sparse-view camera pose estimation compared with naively applying standard flow matching to unconstrained 4D quaternion vectors. QuacamFM also generalizes well across different datasets and scene distributions.

Limitations and future work. While our method explicitly enforces rotation validity during learning, it has not incorporated epipolar geometric constraints due to the expensive correspondence extraction and optimization processes. Future work could integrate epipolar geometry directly into camera pose inference for more accurate estimations, while maintaining low computational overhead.

REFERENCES

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.

Samarth Brahmbhatt, Jinwei Gu, Kihwan Kim, James Hays, and Jan Kautz. Geometry-aware learning of maps for camera localization. In *IEEE/CVF conference on Computer Vision and Pattern Recognition*, pp. 2616–2625, 2018.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *IEEE/CVF International Conference on Computer Vision*, pp. 9630–9640, 2021.

Zhaopeng Cui and Ping Tan. Global structure-from-motion by similarity averaging. In *IEEE International Conference on Computer Vision*, pp. 864–872, 2015.

Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 337–33712, 2018.

Bardienus Pieter Duisterhof, Lojze Zust, Philippe Weinzaepfel, Vincent Leroy, Yohann Cabon, and Jerome Revaud. MASt3r-sfm: a fully-integrated solution for unconstrained structure-from-motion. In *International Conference on 3D Vision*, 2025. URL https://openreview.net/forum?id=5uw1GRBFoT.

Fletcher Dunn and Ian Parbery. *3D Math Primer for Graphics and Game Development*. Taylor & Francis Inc., 2011.

Johan Edstedt, Georg Bökman, Mårten Wadenbäck, and Michael Felsberg. Dedode: Detect, don't describe—describe, don't detect for local feature matching. In *International Conference on 3D Vision*, pp. 148–157. IEEE, 2024a.

Johan Edstedt, Qiyu Sun, Georg Bökman, Mårten Wadenbäck, and Michael Felsberg. Roma: Robust dense feature matching. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19790–19800, 2024b.

Sven Elflein, Qunjie Zhou, and Laura Leal-Taixé. Light3r-sfm: Towards feed-forward structure-from-motion. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16774–16784, 2025.

Michał Gołąbek, Michał Welcer, Cezary Szczepański, Mariusz Krawczyk, Albert Zajdel, and Krystian Borodacz. Quaternion attitude control system of highly maneuverable aircraft. *Electronics*, 11(22), 2022. ISSN 2079-9292. doi: 10.3390/electronics11223775.

Richard Hartley, Andrew Zisserman, et al. *Multiple view geometry in computer vision*. Cambridge university press Cambridge, 2003.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Yuhe Jin, Dmytro Mishkin, Anastasiia Mishchuk, Jiri Matas, Pascal Fua, Kwang Moo Yi, and Eduard Trulls. Image matching across wide baselines: From paper to practice. *International Journal of Computer Vision*, 129(2):517–547, 2021.

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

Jack B Kuipers. *Quaternions and rotation sequences: a primer with applications to orbits, aerospace, and virtual reality*. Princeton university press, 1999.

Cuong Le, Pavlo Melnyk, Urs Waldmann, Mårten Wadenbäck, and Bastian Wandt. Quamo: Quaternion motions for vision-based 3d human kinematics capture. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=em0jPLYjaS.

Vincent Leroy, Yohann Cabon, and Jerome Revaud. Grounding image matching in 3d with mast3r, 2024.

Amy Lin, Jason Y Zhang, Deva Ramanan, and Shubham Tulsiani. Relpose++: Recovering 6d poses from sparse-view observations. In *International Conference on 3D Vision*, pp. 106–115. IEEE, 2024.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, pp. 740–755. Springer, 2014.

Philipp Lindenberger, Paul-Edouard Sarlin, Viktor Larsson, and Marc Pollefeys. Pixel-perfect structure-from-motion with featuremetric refinement. In *IEEE/CVF International Conference on Computer Vision*, pp. 5967–5977, 2021.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling, 2023. URL https://arxiv.org/abs/2210.02747.

Pierre Moulon, Pascal Monasse, and Renaud Marlet. Global fusion of relative motions for robust, accurate and scalable structure from motion. In *IEEE International Conference on Computer Vision*, pp. 3248–3255, 2013.

Raul Mur-Artal and Juan D. Tardos. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE Transactions on Robotics*, 33(5):1255–1262, 2017. doi: 10.1109/TRO.2017.2705103.

Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. Orb-slam: A versatile and accurate monocular slam system. *IEEE transactions on robotics*, 31(5):1147–1163, 2015.

David Nordström, Johan Edstedt, Georg Bökman, Jonathan Astermark, Anders Heyden, Viktor Larsson, Mårten Wadenbäck, Michael Felsberg, and Fredrik Kahl. Loma: Local feature matching revisited. In *European Conference on Computer Vision*, pp. 637–655. Springer, 2026.

Onur Özyeşil, Vladislav Voroninski, Ronen Basri, and Amit Singer. A survey of structure from motion*. *Acta Numerica*, 26:305–364, 2017.

Jeremy Reizenstein, Roman Shapovalov, Philipp Henzler, Luca Sbordone, Patrick Labatut, and David Novotny. Common objects in 3d: Large-scale learning and evaluation of real-life 3d category reconstruction. In *IEEE/CVF International Conference on Computer Vision*, pp. 10881–10891. IEEE, 2021.

Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. From coarse to fine: Robust hierarchical localization at large scale. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12708–12717, 2019.

Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4937–4946, 2020.

Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4104–4113, 2016.

Yoli Shavit, Ron Ferens, and Yosi Keller. Learning multi-scene absolute pose regression with transformers. In *IEEE/CVF International Conference on Computer Vision*, pp. 2713–2722. IEEE, 2021.

Samarth Sinha, Jason Y Zhang, Andrea Tagliasacchi, Igor Gilitschenski, and David B Lindell. Sparsepose: Sparse-view camera pose regression and refinement. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21349–21359, 2023.

Noah Snavely, Steven M Seitz, and Richard Szeliski. Photo tourism: exploring photo collections in 3d. *ACM Transactions on Graphics*, 25(3):835–846, 2006.

Joan Sola. Quaternion kinematics for the error-state kalman filter. *arXiv preprint arXiv:1711.02508*, 2017.

Alexander Tong, Kilian Fatras, Nikolay Malkin, Guillaume Huguet, Yanlei Zhang, Jarrid Rector-Brooks, Guy Wolf, and Yoshua Bengio. Improving and generalizing flow-based generative models with minibatch optimal transport. *arXiv preprint arXiv:2302.00482*, 2023.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Jianyuan Wang, Christian Rupprecht, and David Novotny. Posediffusion: Solving pose estimation via diffusion-aided bundle adjustment. In *IEEE/CVF International Conference on Computer Vision*, pp. 9739–9749, 2023.

Jianyuan Wang, Nikita Karaev, Christian Rupprecht, and David Novotny. Vggsfm: Visual geometry grounded deep structure from motion. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21686–21697, 2024a.

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5294–5306, 2025.

Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20697–20709, 2024b.

Olivia Wiles, Georgia Gkioxari, Richard Szeliski, and Justin Johnson. Synsin: End-to-end view synthesis from a single image. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7465–7475, 2020.

Kyle Wilson and Noah Snavely. Robust global translations with 1dsfm. In *European conference on computer vision*, pp. 61–75. Springer, 2014.

Changchang Wu. Towards linear-time incremental structure from motion. In *International Conference on 3D Vision*, pp. 127–134. IEEE, 2013.

Angxiao Yue, Zichong Wang, and Hongteng Xu. Reqflow: Rectified quaternion flow for efficient and high-quality protein backbone generation. *arXiv preprint arXiv:2502.14637*, 2025.

Jason 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*, volume 2024, pp. 23345–23366, 2024.

Jason Y Zhang, Deva Ramanan, and Shubham Tulsiani. Relpose: Predicting probabilistic relative rotation for single objects in the wild. In *IEEE European Conference on Computer Vision*, pp. 592–611, 2022.

Shangzhan Zhang, Jianyuan Wang, Yinghao Xu, Nan Xue, Christian Rupprecht, Xiaowei Zhou, Yujun Shen, and Gordon Wetzstein. Flare: Feed-forward geometry, appearance and camera estimation from uncalibrated sparse views. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21936–21947, 2025a.

Yichi Zhang, Yici Yan, Alex Schwing, and Zhizhen Zhao. Towards hierarchical rectified flow. In *International Conference on Learning Representations*, pp. 41537–41558, 2025b.

Qitao Zhao, Amy Lin, Jeff Tan, Jason Y Zhang, Deva Ramanan, and Shubham Tulsiani. DiffusionSfM: Predicting structure and motion via ray origin and endpoint diffusion. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6317–6326, 2025.

Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. *arXiv preprint arXiv:1805.09817*, 2018.

APPENDIX

We provide supplementary videos visualizing the evolution of the estimated camera poses throughout the inference process, from their initial states toward the final predictions. These videos show QuacamFM, QuacamFM-`lerp`, and PoseDiffusion. These visualizations can be accessed through the provided anonymous local webpage, `quacamfm_appendix.html`.

Qualitative results on RealEstate10K. We provide qualitative examples of QuacamFM on the RealEstate10K dataset in Fig. 7. The examples cover diverse camera motions, including forward motion combined with rotation and side-to-side motion.

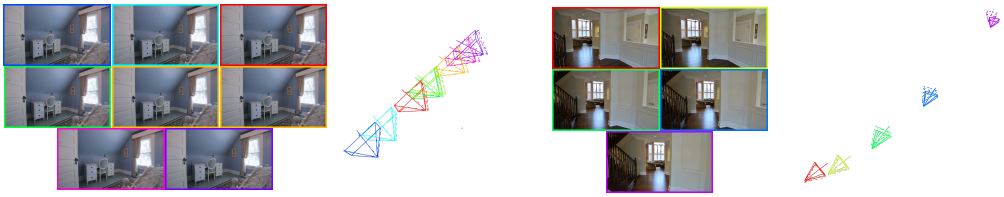

Figure 7: Qualitative results of QuacamFM on RealEstate10K.

Ablation study. We investigate the effect of different visual feature extractors, as in Table 4, by evaluating three backbone configurations: ResNet50, trained by supervised image classification (He et al., 2016); ResNet50-DINO (Caron et al., 2021), trained in a self-supervised manner; and the ViT-DINO (Caron et al., 2021) model. We ablate the number of integration steps used to solve the flow-matching ODE during inference as in Table 5. All ablation experiments are conducted on CO3Dv2.

Table 4: QuacamFM performance with different visual extractor backbones, evaluated with 10-image input.

Backbones	ResNet50	ResNet50-DINO	ViT-DINO
RTA@15	44.1	43.6	**45.1**
RRA@15	77.9	78.4	**82.0**
AUC@30	62.7	64.2	**67.3**

Table 5: QuacamFM performance with different number of timesteps, using the ViT-DINO feature backbone.

Metrics	20 steps	30 steps	50 steps	80 steps
RTA@15	43.9	**45.1**	43.6	44.2
RRA@15	79.5	**82.0**	81.0	79.8
AUC@30	64.2	**67.3**	65.8	66.4

Computational complexity. In Table 6, we report the inference time of QuacamFM, PoseDiffusion w/o GGS, PoseDiffusion, and RelPose++ with different numbers of input images, *e.g.* 5, 10, 20 images. It is measured on a single NVIDIA A100-SXM4-40GB GPU. This shows how the computation of PoseDiffusion and RelPose++ grows when they incorporate epipolar constraint-based optimization and camera rotation consistency optimization. We configure PoseDiffusion to apply the standard GGS during the last 10 denoising steps, and RelPose++ uses its commonly adopted setting of 50 optimization iterations.

Table 6: Inference time of methods with multi-image inputs. The inference time is measured in second (s).

Method	5 images	10 images	20 images
PoseDiffusion	6.344	8.834	20.61
PoseDiffusion w/o GGS	0.342	0.393	0.825
RelPose++	2.854	4.491	11.32
QuacamFM	0.352	0.465	0.735