PosePlaner: Denoising Any Feed-Forward Pose Predictor from Pairwise Planar Geometry

Yihang Chen[1,2] **Bin Tan**[2,‡] **Yujun Shen**[2] **Yinghao Xu**[1,‡] **Nan Xue**[2]

[1]HKUST, Hong Kong, China [2]Robbyant, Ant Group, China

[‡]Corresponding Authors

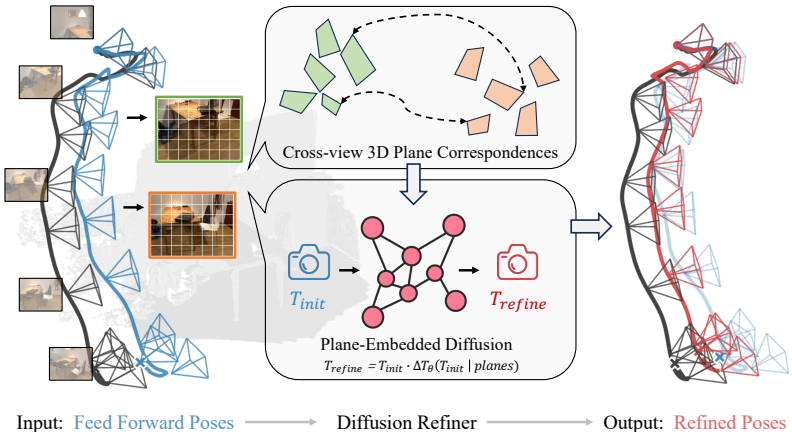


Figure 1: Our proposed POSEPLANER refines any feed-forward pose in one Euler step of a plane-embedded diffusion adapter, and extends to n-view streaming via per-edge view-graph refinement aggregated by rotation averaging and a GLOMAP-style positioner. **Left:** T_{init} (**blue**) drifts from GT (**black**). **Middle:** conditioned on per-view planar primitives, the refiner yields $T_{\text{refine}} = T_{\text{init}} \cdot \Delta T_\theta(T_{\text{init}} \mid \text{planes})$ (**red**) in one step. **Right:** refined trajectory (**red**), raw feed-forward (**blue**), GT (**black**).

Abstract

Recent multi-view Transformers (e.g., DUSt3R and VGGT) have advanced 3D reconstruction by scaling training data and model capacity from two views to many views and long sequences, subsuming the classical SfM/SLAM pipeline of feature extraction, matching, and pose estimation in a single feed-forward pass. Yet this subsumption hides the interfaces between pipeline stages—correspondences, geometry, and pose are entangled in a single forward pass, **leaving supervised fine-tuning as the major path** to improvement and offering no mechanism to correct predictions from geometric evidence at test time. We propose POSEPLANER to close this gap by reviving the classical coupling between correspondences and pose via flow matching, treating pose refinement as a conditional generation problem that takes coarse geometric matches and initial pose as input condition and fuses the resulting pose hypotheses into a single refined estimate, providing a continuous, data-driven analog of RANSAC voting. Our method is trained efficiently on two-view data, yet serves as a periodic refiner for streaming n-view pose estimators with no retraining. Empirically, it corrects failures of feed-forward predictors, substantially improves correspondence-based solvers, and reduces accumulated drift in long streaming sequences. Project page: `https://robbyant-research.github.io/PosePlaner/`.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

1 Introduction

Recovering relative camera pose between views of a scene is a foundational primitive in 3-D vision and the entry point of structure-from-motion [36], multi-view stereo [52], SLAM [30], and novel-view synthesis [29, 17]. The problem is hard at both ends: in the two-view setting under limited baselines and outlier-prone correspondences, and across long streaming sequences where errors accumulate frame by frame. Every pose failure propagates into the downstream pipeline.

Two prominent paradigms in recent work reach this problem from opposite directions. *Feed-forward pose predictors* [46, 49] map an image pair directly to $(R, \boldsymbol{t})$ via visual priors, but their prediction is terminal: downstream correspondences cannot refine it. *Matcher-plus-solver pipelines* [34, 39, 11, 19] hand rich correspondences to a five-point solver [31] wrapped in RANSAC [12], but the matcher's priors collapse into per-pair inlier / outlier decisions and the RANSAC threshold tuning seldom transfers across scenes. Both paradigms break the match-pose coupling: visual priors and per-match geometric evidence never reach the same differentiable channel.

In this paper, we bridge the gap between these two paradigms by reintroducing visual correspondences into feed-forward 3D reconstruction. Analogous to the algebraic solvers that seed the classical SfM pipeline [36], we treat feed-forward models as pose initializers and refine their output as a learned geometric correction step. Our key insight is that 3D surface points paired with their normals already encode a cross-view geometric constraint: under the true relative pose, matched patch-normal pairs must satisfy an algebraic warping identity, and departures from it yield differentiable residuals that grow with pose error. Planar primitives realize this constraint compactly—one 3-D center and one normal per image patch. We learn to denoise the pose residual via flow matching, conditioned on the initialized pose and matched patch-normal pairs, yielding a refiner that generalizes from two-view training to N-view inference without retraining.

We introduce POSEPLANER (Fig. 1), a pose refiner that accepts any initial pose T_{init} and returns a refined relative pose given patch-normal matching pairs. Building on PLANA3R [23], we use planar primitives as compact patch-normal representations, where each patch predicts a 3D center and plane normal in its own camera frame; pairing these patch-level primitives via an external pixel matcher (e.g., MAST3R [19], LOFTR [39]) yields M two-view correspondences whose warping residuals serve as the conditioning signal for the refiner. For pose refinement in SE(3), we exploit the asymmetric difficulty of rotation and translation: a flow-matching [27, 22] multi-query Transformer [43, 33] denoises the rotation residual conditioned on the plane-warping residuals, producing M hypotheses that a learned score head fuses with the initial pose by softmax-weighted voting; translation is then recovered by a geometric solver under the refined rotation. Trained once on two-view pairs, POSEPLANER extends as a periodic tail refiner for streaming n-view pose estimators on long sequences with no retraining. Beyond classical bundle adjustment and pose-graph backends [36], to the best of our knowledge this is the first *learned, diffusion-based* refiner that operates on the pose outputs of streaming n-view estimators without sequence-specific retraining. Experimentally, our POSEPLANER consistently improves upstream estimators from both paradigms on two benchmarks, and its gains grow with sequence length in the streaming setting, where drift correction matters most—e.g., on ScanNet at $N{=}2000$, rotation AUC@5° improves by $+20$ points for CUT3R [47] and $+21$ for TTT3R [7].

Our contributions.

- We propose POSEPLANER, the first *learned, diffusion-based* pose refiner that revives the classical correspondence–pose coupling for feed-forward 3D reconstruction models, treating pose refinement as a conditional generation problem over plane-warping residuals.
- We design a hybrid SE(3) refinement scheme: a flow-matching multi-query Transformer for rotation with score-based hypothesis fusion, and a geometric solver for translation, enabling single-step inference.
- We show that a model trained once on two-view pairs serves as a periodic tail refiner for streaming n-view estimators with no retraining.

2 Related Work

Two-view and feed-forward pose estimation. Classical structure-from-motion [36, 30] recovers relative pose from sparse correspondences via a closed-form essential-matrix solver [31] wrapped in

RANSAC [12]; modern matchers [34, 39, 11, 19] replace hand-crafted descriptors with learned dense features, but the discrete solver still collapses each match into an inlier / outlier vote, and a single bad RANSAC threshold can erase the entire inlier set. Differentiable RANSAC variants [3, 1, 2] relax this voting into a soft, end-to-end-trainable selection, yet gradients still flow through aggregate inlier counts rather than per-correspondence geometric residuals. A complementary line skips the solver and regresses pose directly via large feed-forward 3-D backbones [49, 19, 46] or their online streaming descendants [47, 7, 21, 55, 60, 18, 44, 6]; both produce a terminal prediction whose downstream correspondences and accumulated drift never re-enter the prediction loop. POSEPLANER sits downstream of either paradigm, correcting any initial pose using the geometric constraint encoded in matched plane primitives and restoring the pose–correspondence coupling that both break.

Planar primitives for 3-D understanding. Indoor scenes are dominated by piecewise-planar structure, motivating reconstruction by sparse 3-D plane primitives rather than dense points or meshes [54, 16, 50, 41, 58, 26, 23]. Early feed-forward methods [24, 25, 37] regress plane masks and per-pixel plane parameters via instance segmentation, but depend on manually annotated plane labels (SCANNETV2 [8], MATTERPORT3D [4]), capping both training scale and cross-dataset generalization. PLANARSPLATTING [41] breaks this bottleneck with a differentiable plane-primitive rasterizer whose rendering loss is annotation-free, and PLANA3R [23] uses it to train a feed-forward stereo plane-primitive backbone at scale, predicting per-patch normals and centers end-to-end. On the pose side, NOPESAC [40] matches a few scene-level plane instances, feeds each pair through a shared MLP to produce one pose hypothesis per correspondence, and soft-fuses these (with the init) via a learned scoring layer over their warping costs. POSEPLANER retains the soft-fusion idea but operates at the much denser *patch level* and replaces the per-plane MLP regressor with a *continuous flow-matching denoiser* jointly conditioned on the plane-warping residuals of all matched patch pairs.

Diffusion models for pose and 3-D. Denoising diffusion [14] has been adapted to pose by POSEDIFFUSION [45] (camera trajectories conditioned on image features), RAYDIFFUSION [56] (denoising a per-pixel ray bundle), and DIFFUSIONSFM [57] (jointly denoising per-pixel ray origins and endpoints to recover structure and motion); all three are visual-prior-driven, with no explicit per-correspondence geometric channel. Flow matching [22, 27] replaces the noise-prediction objective with a deterministic vector-field target, enabling single-step inference. POSEPLANER adopts this flow-matching parameterization and conditions on *plane-warping residuals* rather than pixels, so the denoiser sees the algebraic quantity a classical solver would minimize—a geometric channel decoupled from any single matcher's visual statistics.

3 Method

POSEPLANER takes any initial two-view relative pose T_{init} of the input image pair $(\mathcal{I}_1, \mathcal{I}_2)$ and refines it into a more accurate estimate. The main idea rests on a classical observation:

Proposition 1 (Plane-Warping Identity)**.** *Let $(\boldsymbol{n}_1, d_1)$ and $(\boldsymbol{n}_2, d_2)$ be the parameters of a 3D plane observed in two camera frames related by a relative pose $T_{rel} = (R, \boldsymbol{t})$. Then*

$$\boldsymbol{n}_1 = R\,\boldsymbol{n}_2, \qquad d_1 = d_2 - (R\,\boldsymbol{n}_2)^\top \boldsymbol{t}. \tag{1}$$

Departures from this identity produce the plane-warping residuals

$$\Delta\boldsymbol{n}(R) = \|\boldsymbol{n}_1 - R\,\boldsymbol{n}_2\|_2\,, \qquad \Delta d(R, \boldsymbol{t}) = |d_1 - d_2 + (R\,\boldsymbol{n}_2)^\top \boldsymbol{t}|, \tag{2}$$

which vanish at the true pose and grow with pose error.

These residuals provide an estimator-agnostic conditioning signal for any upstream source. Given a two-view Vision Transformer that reads out patch-level 3D planes in each view's own camera frame (Sec. 3.1), we compute the plane-warping residuals between matched pairs and use them as conditioning for a flow-matching Transformer that denoises the rotation residual; translation is then recovered by a geometric solver under the refined rotation (Sec. 3.2). See Fig. 2 for an overview.

3.1 Patchwise Plane Primitives and Correspondences

Our method follows the convention of Vision Transformers [10], patchifying the input images $(\mathcal{I}_1, \mathcal{I}_2)$ into S patch tokens each. A siamese ViT encoder processes both views independently, and two

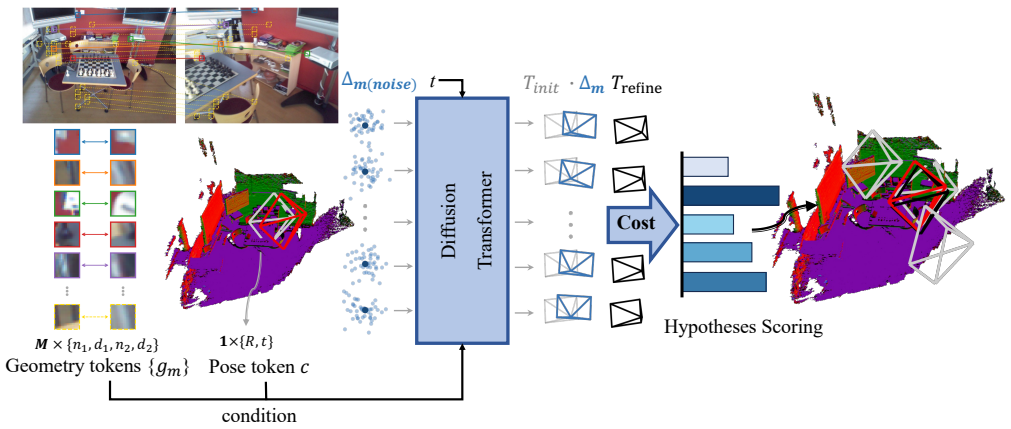


Figure 2: **POSEPLANER pipeline.** A per-view local-frame plane head produces M geometry tokens $\{\boldsymbol{g}_m\}$ (paired plane primitives across the two views) and a global pose token $\boldsymbol{c}$ that encodes the initial relative pose T_{init}. The multi-query diffusion Transformer denoises T_{init} in a single Euler step under these conditioning tokens, yielding M hypothesis frustums $h_1, \ldots, h_M$. A score head softly fuses the hypotheses into the final refined pose T_{refined}, used to align the two-view reconstruction.

cross-attention decoders fuse information across views, predicting a 3D planar primitive per patch, specifically a center depth, two radii, and an orientation quaternion from which the plane normal $\boldsymbol{n}$ and offset d are recovered.

Plane representation from Siamese Vision Transformers. Each patch in a view predicts a local planar primitive parameterized by a 3D center $\boldsymbol{P} \in \mathbb{R}^3$ and a unit surface normal $\boldsymbol{n} \in S^2$, both expressed in its own camera frame. The plane offset follows as $d = -\boldsymbol{n}^\top \boldsymbol{P}$, giving a compact $(\boldsymbol{n}, d)$ per patch. We adopt the architecture of PLANA3R [23] with one key change: whereas PLANA3R expresses all primitives in a shared global frame (view 1), we set each primitive to live in its own source camera's frame. The training details of this Siamese Vision Transformer network are in App. B.1.

Two-view plane correspondences. After the plane head produces $(\boldsymbol{n}, d)$ for every patch in both views, we establish patch-level correspondences from any external pixel matcher (e.g., SuperGlue [34], LoFTR [39], ROMA [11], or MASt3R [19]). Given pixel-level matches, we convert each matched pixel pair to its enclosing patch indices and accumulate a vote matrix $V \in \mathbb{R}^{S \times S}$ counting the number of pixel matches per patch pair. Patch-level correspondences are then extracted by mutual nearest neighbors: patch i in view 1 is matched to patch j in view 2 if and only if $j = \arg\max_k V_{ik}$ and $i = \arg\max_k V_{kj}$. This yields M matched plane pairs $\{(\boldsymbol{n}_m^1, d_m^1), (\boldsymbol{n}_m^2, d_m^2)\}_{m=1}^M$, which we pack into geometry tokens

$$\boldsymbol{g}_m \;=\; \left[\, \boldsymbol{n}_m^1 \,\|\, \bar{d}_m^1 \,\|\, \boldsymbol{n}_m^2 \,\|\, \bar{d}_m^2 \,\right] \in \mathbb{R}^8, \qquad m = 1, \ldots, M, \tag{3}$$

where $\bar{d}_m^v$ denotes the scene-normalized offset (Sec. 3.2). Each token is the minimal sufficient statistic for evaluating the plane-warping residuals of Eq. (2) at any candidate pose. Note that M is not a fixed hyperparameter: it is determined per image pair by the matcher's mutual-NN output and varies with scene texture and matcher density (typically tens to a few hundred per pair for the matchers we use). The M residuals collectively expose the pose error along diverse normal orientations, forming the conditioning signal consumed by the flow-matching refiner below.

3.2 Flow-Matching Diffusion Refiner

The refiner takes $T_{\text{init}} = (R_{\text{init}}, \boldsymbol{t}_{\text{init}})$ and the geometry tokens $\{\boldsymbol{g}_m\}$ of Sec. 3.1, and produces a refined pose via flow matching [22, 27]. For $t \sim \mathcal{U}[0,1]$ and $\boldsymbol{\epsilon} \sim \mathcal{N}(0, I)$, the formulation defines a linear probability path and a constant target velocity

$$\boldsymbol{z}_t \;=\; (1-t)\,\boldsymbol{z}_0 + t\,\boldsymbol{\epsilon}, \qquad \boldsymbol{v}^\star(\boldsymbol{z}_t, t) \;=\; \boldsymbol{\epsilon} - \boldsymbol{z}_0, \tag{4}$$

and the refiner F_θ regresses $\boldsymbol{v}^\star$ at every $(\boldsymbol{z}_t, t)$. The straight-line path admits a *single* deterministic Euler step at inference, so refinement reduces to one forward pass (see App. A.3 for the formal argument).

Velocity field parameterization. The velocity field F_θ is specified by three choices: the coordinate system, the denoising target, and the conditioning signal.

We normalize all plane offsets by a per-pair scalar $\sigma = \frac{1}{2}(\overline{|d^1|} + \overline{|d^2|})$ so that $\bar{d}^v_m = d^v_m/\sigma$. A robust median over the matched pairs recovers the normalized translation scale:

$$s_{\text{est}}(R, \hat{\boldsymbol{t}}) \;=\; \text{median}_{m\in\mathcal{M}^\star}\left|\frac{\bar{d}^2_m - \bar{d}^1_m}{(R\boldsymbol{n}^2_m)^\top \hat{\boldsymbol{t}}}\right|, \qquad \boldsymbol{\tau} \;=\; s_{\text{est}}(R, \hat{\boldsymbol{t}})\,\hat{\boldsymbol{t}}, \tag{5}$$

where $\mathcal{M}^\star$ excludes matches with near-zero denominators. This maps any T_{init} into the range of $\bar{d}$ regardless of its metric scale; the metric translation is recovered by $\boldsymbol{t} = \sigma\,\boldsymbol{\tau}$.

The denoising target is the 9-dimensional residual

$$\boldsymbol{z}_0 \;=\; \left[\; \text{rot6d}(R_{\text{gt}}R_{\text{init}}^\top) \;\middle\|\; \boldsymbol{\tau}_{\text{gt}} - \boldsymbol{\tau}_{\text{init}} \;\right] \in \mathbb{R}^9, \tag{6}$$

with rotation in the continuous 6-D representation of Zhou et al. [59] (see App. A.2 for why quaternions are suboptimal under flow matching). Although our final output uses only the rotation, the translation channel acts as a contextual signal that tightens the rotation prediction (Sec. 4.3).

The refiner conditions on the M geometry tokens $\{\boldsymbol{g}_m\}$ from Eq. (3) and a global pose token $\boldsymbol{c} = \psi([\text{rot6d}(R_{\text{init}})\|\boldsymbol{\tau}_{\text{init}}]) \in \mathbb{R}^d$. Linear projections lift both to width d and $L_{\text{cond}}{=}4$ self-attention blocks produce the context $\hat{\boldsymbol{C}}$ consumed by the denoiser.

Multi-query denoiser. The denoiser instantiates M queries, one per correspondence, each solving the same pose-residual denoising problem under a different local geometric condition. Every query shares a latent-time component

$$\boldsymbol{q}_0 \;=\; \phi_z(\boldsymbol{z}_t) + \phi_t\big(\text{SinEmb}(t)\big) \in \mathbb{R}^d, \tag{7}$$

and adds a per-match geometry bias $\boldsymbol{q}_m = \boldsymbol{q}_0 + W_b\,\hat{\boldsymbol{g}}_m$ that grounds each query in correspondence m. The M queries cross-attend to $\hat{\boldsymbol{C}}$ and emit per-match velocity predictions $\boldsymbol{v}_m \in \mathbb{R}^9$. During training, these per-match predictions are aggregated into a single fused velocity by geometric weights

$$\omega_m \;=\; \text{softmax}(-\beta\, g_m), \qquad g_m \;=\; \|\Delta\boldsymbol{n}_m\|_2 + \lambda\,|\Delta d_m|, \qquad (\lambda, \beta) = (3, 10), \tag{8}$$

so the M queries cooperatively regress the shared FM target $\boldsymbol{v}^\star$ via their weighted average; correspondences with small warping residuals dominate the gradient. Per-query diversity at inference arises from the per-match geometry bias $W_b\hat{\boldsymbol{g}}_m$, which biases each $\boldsymbol{v}_m$ toward a different local geometric condition. At inference, each query is integrated independently into one of M hypothesis poses, and the $M{+}1$ hypotheses (with T_{init}) are fused by the score head below.

Score-based fusion. A score head evaluates each of the $M{+}1$ hypotheses (including T_{init}) on plane-warping costs

$$c^r_{h,m} = \big\|R_h\boldsymbol{n}^2_m - \boldsymbol{n}^1_m\big\|_2, \qquad c^t_{h,m} = \big\|(\bar{d}^2_m - (R_h\boldsymbol{n}^2_m)^\top\boldsymbol{\tau}_h)\,R_h\boldsymbol{n}^2_m - \bar{d}^1_m\boldsymbol{n}^1_m\big\|_2, \tag{9}$$

and produces softmax weights $\boldsymbol{w}_r, \boldsymbol{w}_t$ over the hypotheses. The fused refined pose reads

$$R_{\text{ref}} = \Pi_{\text{SO(3)}}\Big(\sum_{h=0}^{M} w_{r,h}\;\text{rot6d}(R_h)\Big), \qquad \boldsymbol{\tau}_{\text{ref}} = \sum_{h=0}^{M} w_{t,h}\,\boldsymbol{\tau}_h, \tag{10}$$

with $\Pi_{\text{SO(3)}}$ Gram–Schmidt orthogonalization.

Training objective. The refiner and the score head are optimized jointly with

$$\mathcal{L}_{\text{refine}} \;=\; \underbrace{\Big\|\sum_m \omega_m\,\boldsymbol{v}_m - \boldsymbol{v}^\star\Big\|_1}_{\mathcal{L}_{\text{FM}}} + 0.1\cdot\underbrace{\mathbb{E}_t\Big[(1{-}t)^2\sum_m \omega_m\,g_m\Big]}_{\mathcal{L}_{\text{geo}}} + 0.01\cdot\mathcal{L}_{\text{score}}, \tag{11}$$

where $\mathcal{L}_{\text{score}}$ denotes the hypothesis-selection loss of [40], concentrating $\boldsymbol{w}_r$ (resp. $\boldsymbol{w}_t$) on the hypothesis with the smallest rotation (resp. translation) error and penalizing off-diagonal entries of the correspondence-indexed cost matrix. The three terms encode complementary signals: $\mathcal{L}_{\text{FM}}$ drives the velocity regression on the pose residual, $\mathcal{L}_{\text{geo}}$ constrains the predicted trajectory to the $(\Delta\boldsymbol{n}, \Delta d)\to\mathbf{0}$ manifold of Eq. (2), and $\mathcal{L}_{\text{score}}$ supervises the soft hypothesis selection used in Eq. (10). The temporal weight $(1{-}t)^2$ in $\mathcal{L}_{\text{geo}}$ concentrates geometric supervision on iterates near $\boldsymbol{z}_0$, where the warping residuals become linearly informative of the remaining pose error.

3.3 Inference

At test time we sample $\boldsymbol{z}_1 \sim \mathcal{N}(0, I)$, broadcast it to the M queries, and take a *single* forward Euler step *per query* along the learned velocity field,

$$\boldsymbol{z}_{0,m} \;=\; \boldsymbol{z}_1 - \boldsymbol{v}_m(\boldsymbol{z}_1,\, t{=}1), \qquad m = 1, \ldots, M, \tag{12}$$

whose unit step covers the full $t \in [0, 1]$ interval. Decoding each $\boldsymbol{z}_{0,m}$ and composing with T_{init} yields M pose hypotheses $\{T_m\}_{m=1}^{M}$ in one evaluation of F_θ; the score head then fuses them with T_{init} through Eq. (10) into the refined pose $(R_{\text{ref}}, \boldsymbol{\tau}_{\text{ref}})$. We take $R^\star = R_{\text{ref}}$ as the final rotation and recover the metric translation by the geometric step below, warm-started from $\boldsymbol{t}_{\text{init}}$.

Two-view translation refinement. With R fixed to $R^\star$, translation reduces to a near-linear problem. We lock the magnitude $s = \|\boldsymbol{t}_{\text{init}}\|$ at the scale from Eq. (5) and optimize only the unit direction $\hat{\boldsymbol{t}} \in S^2$ via a Huber-robust objective combining the plane-offset residual Δd_m and a 3-D point-alignment residual $\Delta \boldsymbol{P}_m = \boldsymbol{P}_m^1 - R^\star \boldsymbol{P}_m^2 - s\hat{\boldsymbol{t}}$, where $\boldsymbol{P}_m^v$ is the back-projection of matched pixel m in view v through the dense depth that PLANARSPLATTING rasterizes from the plane-head primitives. The objective is regularized by an init-anchored Tikhonov prior on the direction:

$$\hat{\boldsymbol{t}}^\star \;=\; \operatorname*{arg\,min}_{\hat{\boldsymbol{t}} \in S^2} \sum_{m \in \mathcal{M}} \Big[\rho_d\big(\Delta d_m\big) + \rho_p\big(\|\Delta \boldsymbol{P}_m\|_2\big) \Big] \;+\; \lambda_{\text{reg}}\, \|\hat{\boldsymbol{t}} - \hat{\boldsymbol{t}}_{\text{init}}\|_2^2. \tag{13}$$

ρ_d and ρ_p are Huber loss functions and λ_{reg} is set adaptively from the init's self-consistency with the predicted plane field, so that an already-accurate init is protected (large λ_{reg}, solver returns $\approx \hat{\boldsymbol{t}}_{\text{init}}$) while a poor init lets the data terms dominate. The closed-form schedule and the construction of the dense $\boldsymbol{P}_m^v$ are detailed in App. B.3.

Multi-view aggregation. For streaming n-view inputs, we run the diffusion refiner on every edge of the refinement window to obtain refined pairwise rotations $\{R_{ij}^\star\}$ while keeping the upstream's raw pairwise translation directions $\{\boldsymbol{t}_{ij}^{\text{raw}}\}$ unchanged. Per-frame absolute rotations are then recovered by rotation averaging [13, 5] on $\{R_{ij}^\star\}$, and per-frame camera centers by a GLOMAP-style fix-rotation positioner [32] initialised from the raw translations; the explicit objectives are given in App. B.3. The two-view LM translation solver of Eq. (13) is bypassed at n-view scale—per-frame translations are determined globally by RA + the positioner from the dense edge set—so the same learned refiner (whose only role at n-view is to refine each R_{ij}) serves both regimes without retraining.

4 Experiment

We evaluate POSEPLANER as a plug-in refiner for both two-view and streaming n-view pose estimators across two benchmarks, ScanNet1500 [8] and 7Scenes [38], in Sec. 4.1 and Sec. 4.2; an additional zero-shot two-view evaluation on Matterport3D [4] (SPARSEPLANES [16] test split) is reported in App. C.1 (Tab. 7). Implementation and training details are in App. B.

4.1 Two-View Camera Pose Refinement

Evaluation protocol. We evaluate on ScanNet1500 [8] (1500 standard test pairs from [34]) and 7Scenes [38] (4070 pairs formed at frame gap 50 across all 46 sequences; zero-shot, not in our training mixture). We attach POSEPLANER downstream of seven upstream pose sources: four correspondence-based solvers (SUPERGLUE [34], LOFTR [39], ROMA [11], MAST3R [19]) and three diffusion-based pose estimators (POSEDIFFUSION [45], RAYDIFFUSION [56], DIFFUSIONSFM [57]). Each correspondence-based upstream uses its own native pose solver followed by a two-view bundle adjustment to produce the init pose. We report pairwise rotation and translation AUC at 5° and 10° (higher is better).

NopeSAC configuration. We evaluate the frozen NOPESAC [40] refinement head using MAST3R patch matches and our predicted plane pairs. Initial poses are post-BA estimates for correspondence-based upstreams and independently rerun predictions for diffusion-based upstreams on the same test pairs. We bypass its PlaneTR frontend, native matching, and pixel camera prediction, retaining the official plane warp, AIM, 50-slot head, and soft fusion. The released ScanNet checkpoint is used without retraining on either benchmark. We report its directly predicted rotation and translation; POSEPLANER retains its geometric translation refinement (Eq. (13)). This evaluates the released head under shared frontend inputs.

Table 1: **Two-view pose refinement on ScanNet1500 [8] and 7Scenes [38].** R/t: rotation/translation-direction AUC at 5° and 10° (%, higher is better). *Valid* gives the common evaluation subset within each block. POSEPLANER uses MAST3R conditioning throughout (Tab. 9). Best per block in **bold**. † Tested zero-shot.

Method	ScanNet1500					7Scenes†				
	R@5	R@10	t@5	t@10	Valid	R@5	R@10	t@5	t@10	Valid
Correspondence-based solvers										
SUPERGLUE [34]	43.9	62.4	21.3	39.6	1427/1500	42.3	65.9	18.9	37.6	4014/4070
SUPERGLUE + NOPESAC head	34.9	58.7	7.4	19.9		34.3	61.5	7.1	19.9	
SUPERGLUE + POSEPLANER	**48.9**	**68.9**	**21.5**	**40.4**		**45.8**	**69.7**	**19.0**	**37.6**	
LOFTR [39]	11.5	25.0	2.1	5.6	1028/1500	17.7	34.2	3.7	10.1	3585/4070
LOFTR + NOPESAC head	26.7	49.9	1.6	4.6		27.1	52.8	2.6	8.3	
LOFTR + POSEPLANER	**29.4**	**52.9**	**3.5**	**9.0**		**31.2**	**56.8**	**4.4**	**12.7**	
ROMA [11]	**66.7**	**81.2**	**39.1**	**58.8**	1500/1500	50.7	73.3	**25.6**	**46.7**	4068/4070
ROMA + NOPESAC head	38.6	61.8	9.8	25.6		35.4	62.6	8.4	23.2	
ROMA + POSEPLANER	64.1	80.1	37.0	57.3		**52.0**	**74.1**	25.4	46.1	
MAST3R [19]	**66.5**	**82.4**	**41.3**	**61.6**	1495/1500	50.6	73.3	**26.7**	**47.9**	4065/4070
MAST3R + NOPESAC head	38.2	61.6	10.2	26.0		35.4	62.7	8.8	23.2	
MAST3R + POSEPLANER	65.0	81.7	38.6	59.8		**51.5**	**74.0**	26.0	47.0	
Diffusion-based pose estimators (visual conditioning)										
POSEDIFF. [45]	2.2	9.0	0.5	1.2	1500/1500	3.2	12.2	0.4	1.5	4070/4070
POSEDIFF. + NOPESAC head	18.2	**41.3**	0.8	3.0		18.0	42.4	0.4	1.5	
POSEDIFF. + POSEPLANER	**18.9**	38.8	**2.2**	**6.0**		**20.6**	**44.6**	**1.0**	**3.8**	
RAYDIFF. [56]	11.7	33.4	1.7	6.9	1500/1500	10.3	32.7	1.0	4.4	4070/4070
RAYDIFF. + NOPESAC head	6.5	19.2	0.3	1.3		7.4	20.5	0.2	0.7	
RAYDIFF. + POSEPLANER	**25.3**	**50.5**	**5.4**	**15.5**		**23.3**	**51.4**	**2.2**	**7.6**	
DIFFUSIONSFM [57]	37.8	64.3	3.0	16.0	1500/1500	39.6	66.6	3.9	16.5	4070/4070
DIFFUSIONSFM + NOPESAC head	35.0	59.5	4.1	13.8		33.5	61.1	4.3	14.9	
DIFFUSIONSFM + POSEPLANER	**43.6**	**68.6**	**12.8**	**30.5**		**43.6**	**69.1**	**6.2**	**20.3**	

Table 2: **Controlled comparison with bundle adjustment.** BA and POSEPLANER use the same initial poses and MAST3R correspondences. R@5, R@10, and R@20 are rotation AUC at 5°, 10°, and 20° (%, higher is better). Best results in each initialization group are **bold**. Table 1's correspondence-based initializations have already undergone BA.

Initialization / refinement	ScanNet1500			7Scenes		
	R@5	R@10	R@20	R@5	R@10	R@20
LOFTR (pre-BA)	11.5	25.2	44.1	17.8	34.3	53.4
+ BA	13.4	26.5	44.9	24.4	39.6	56.7
+ POSEPLANER	**29.5**	**53.1**	**70.7**	**31.2**	**56.8**	**75.4**
MAST3R (pre-BA)	66.0	82.1	90.9	49.6	72.7	86.1
+ BA	**66.5**	**82.4**	**91.1**	50.6	73.3	86.4
+ POSEPLANER	64.7	81.5	90.6	**50.9**	**73.6**	**86.5**

Quantitative results. In Tab. 1, POSEPLANER outperforms the NOPESAC head at both reported rotation and translation thresholds for all four correspondence-based upstreams on both benchmarks. The visual-conditioning outputs of POSEDIFFUSION, RAYDIFFUSION, and DIFFUSIONSFM also improve substantially after refinement, indicating that the plane-warping residuals carry information complementary to visual-only priors.

Comparison with classical refinement. Both methods start from the same LOFTR- or MAST3R-derived pre-BA poses and share the same MAST3R matching output. BA uses MAST3R's point correspondences, while POSEPLANER uses patch-level correspondences. Overall, POSEPLANER substantially outperforms BA with LOFTR initialization on both datasets and achieves higher rotation AUC with MAST3R initialization on 7Scenes (Tab. 2).

Plane primitives also work as a stand-alone match filter. The per-patch plane primitives also help *without* invoking the refiner: applying the cut $\|\Delta \boldsymbol{n}\| + |\Delta d| \leq 1$ to LOFTR matches removes a

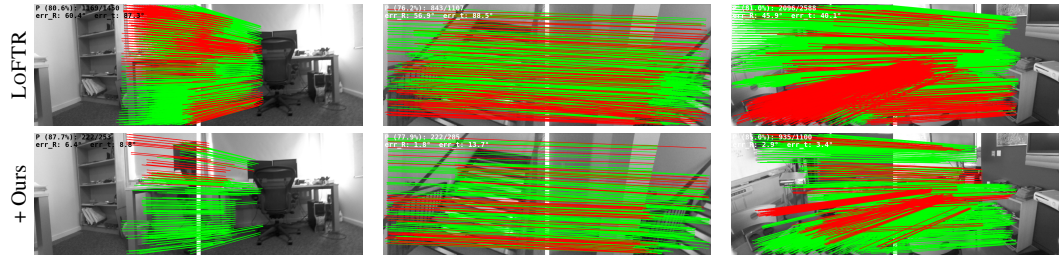


Figure 3: **Planar primitives as a stand-alone match filter.** Three 7SCENES pairs, LOFTR [39] upstream. **Top:** raw LOFTR matches. **Bottom:** matches surviving the cut $\|\Delta \boldsymbol{n}\| + |\Delta d| \leq 1$. **Green**/**red** mark RANSAC inliers/outliers; top-left numbers are inlier ratio and pose error.

Table 3: n**-view pose accuracy on streaming ScanNet [8] and 7Scenes [38].** Pairwise R and t AUC at 5° and 20° over the $\binom{100}{2}$ pairs in the refined tail. Higher is better. † Tested zero-shot.

Method	N=200				N=500				N=2000			
	R@5	R@20	t@5	t@20	R@5	R@20	t@5	t@20	R@5	R@20	t@5	t@20
ScanNet												
CUT3R	23.3	65.0	4.1	24.0	16.5	55.7	2.2	19.0	17.3	53.4	1.0	13.2
CUT3R + POSEPLANER	**40.5**	**80.3**	**7.7**	**38.0**	**37.3**	**78.9**	**6.3**	**38.4**	**37.3**	**79.8**	**3.4**	**31.2**
TTT3R	46.2	83.1	**11.1**	42.7	36.1	79.3	8.0	38.7	27.5	72.5	3.4	28.6
TTT3R + POSEPLANER	**50.7**	**85.6**	10.8	**45.9**	**47.1**	**84.3**	**10.8**	**47.3**	**48.2**	**85.0**	**10.4**	**41.8**
WINT3R	37.7	77.1	8.4	37.4	28.3	66.3	5.5	28.8	19.2	48.6	3.1	16.8
WINT3R + POSEPLANER	**45.9**	**83.7**	**9.5**	**42.7**	**42.5**	**80.7**	**7.8**	**41.3**	**27.9**	**65.0**	**3.9**	**20.8**
SPANN3R	32.4	69.2	6.9	31.9	20.0	55.6	3.5	21.4	19.4	52.8	1.8	16.0
SPANN3R + POSEPLANER	**45.6**	**82.2**	**9.3**	**42.2**	**38.8**	**76.4**	**7.0**	**37.1**	**37.1**	**77.4**	**4.2**	**30.9**
7Scenes†												
CUT3R	22.0	65.4	3.2	24.8	16.2	54.4	2.0	17.4	12.0	47.3	0.4	7.7
CUT3R + POSEPLANER	**35.1**	**77.7**	**5.9**	**34.8**	**35.1**	**78.0**	**4.7**	**35.7**	**28.1**	**71.0**	**2.2**	**26.0**
TTT3R	44.1	**83.9**	**11.2**	**48.8**	36.0	79.4	7.9	40.5	38.4	82.1	3.4	27.1
TTT3R + POSEPLANER	**44.5**	83.3	11.0	47.8	**41.9**	**83.7**	**9.7**	**47.3**	**41.7**	**83.6**	**4.0**	**31.2**
WINT3R	40.0	81.7	7.4	37.8	44.2	82.9	12.7	45.7	41.6	78.7	12.7	35.5
WINT3R + POSEPLANER	**43.9**	**83.4**	**7.6**	**42.4**	**46.3**	**85.0**	**12.8**	**49.4**	**42.9**	**81.2**	**15.2**	**38.4**
SPANN3R	34.5	77.2	7.7	37.3	35.7	76.5	8.1	38.9	39.6	77.0	**9.2**	32.5
SPANN3R + POSEPLANER	**40.8**	**81.8**	**8.9**	**42.9**	**42.1**	**82.6**	**9.4**	**46.8**	**43.7**	**79.7**	8.2	**33.9**

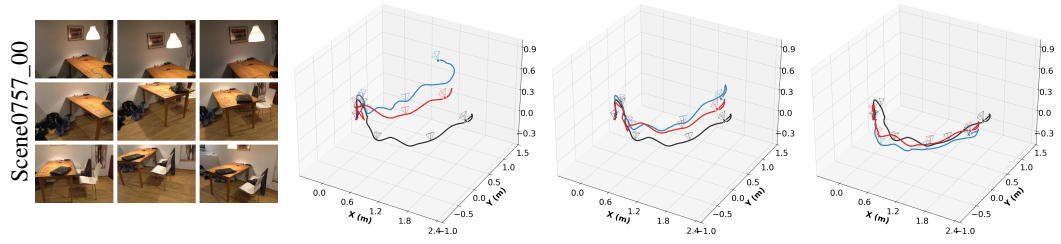


Figure 4: **Streaming pose refinement on a representative ScanNet [8] sequence (Scene0757_00).** Left: sample frames. Right three panels: estimated camera trajectories under CUT3R [47], TTT3R [7], and WINT3R [21]. **Black**: ground-truth; **blue**: raw baseline; **red**: after POSEPLANER refinement. • start, × end.

large fraction of RANSAC outliers while preserving inliers, and re-solving pose on the filtered subset substantially reduces rotation and translation error (Fig. 3).

4.2 Multi-View Refinement on Streaming Pose Estimators

Evaluation protocol. On ScanNet [8] and 7Scenes [38] (zero-shot), we form N-frame trajectories at $N \in \{200, 500, 2000\}$ at frame gap 2; the streaming baseline ingests the first N frames unrefined and POSEPLANER refines the last 100 pairwise. Both raw and refined per-pair $(R, \boldsymbol{t})$ pass through the *same* multi-view backend (rotation averaging + fix-rotation GLOMAP-style positioner [32]; full details in App. B.3), so the only difference is whether each input pair is refined. We evaluate four streaming reconstructors (CUT3R [47], TTT3R [7], WINT3R [21], SPANN3R [44]) and report pairwise rotation and translation AUC at 5° and 20° over the $\binom{100}{2}$ tail pairs.

Table 4: **Zero-shot outdoor multi-view refinement.** R@5/R@20 and t@5/t@20 are rotation and translation-direction AUC at 5°/20° (%, ↑). Better results within each pair are **bold**. † Tested zero-shot.

Method	$N = 200$				$N = 500$				$N = 2000$			
	R@5	R@20	t@5	t@20	R@5	R@20	t@5	t@20	R@5	R@20	t@5	t@20
Oxford Spires†												
CUT3R	59.1	85.1	28.9	66.2	35.5	75.7	8.1	52.5	34.0	70.1	11.6	46.5
CUT3R + POSEPLANER	**65.8**	**90.8**	**33.5**	**70.5**	**50.4**	**82.8**	**22.4**	**64.1**	**53.3**	**85.8**	**22.7**	**67.4**
TTT3R	**73.6**	**92.7**	**45.8**	**77.3**	59.7	88.5	27.5	**74.9**	37.2	76.5	11.4	53.2
TTT3R + POSEPLANER	72.3	92.5	42.0	74.9	**62.9**	**89.2**	**29.8**	74.0	**51.3**	**85.6**	**19.9**	**67.9**
MatrixCity†												
CUT3R	13.2	48.5	4.8	20.1	12.4	49.3	3.1	15.9	9.9	40.4	3.7	15.2
CUT3R + POSEPLANER	**16.4**	**55.6**	**5.6**	**22.8**	**15.1**	**53.5**	**3.5**	**17.8**	**15.5**	**54.0**	**8.2**	**22.2**
TTT3R	18.1	52.7	5.2	18.8	10.5	45.3	4.3	18.7	12.7	47.2	3.8	15.8
TTT3R + POSEPLANER	**20.6**	**56.9**	**6.6**	**23.0**	**14.4**	**49.7**	**5.2**	**19.4**	**17.2**	**54.5**	**7.3**	**20.2**

Table 5: **Ablation study on ScanNet1500** with SUPERGLUE and LOFTR upstream. R and t AUC at $\tau \in \{5, 10, 20\}^\circ$; higher is better, best per column in **bold**. Removing the pose token $\boldsymbol{c}$ forces the model to estimate pose without the upstream init, so the SUPERGLUE and LOFTR blocks of that row are identical by construction.

Variant	Init: SUPERGLUE						Init: LOFTR					
	R@5	R@10	R@20	t@5	t@10	t@20	R@5	R@10	R@20	t@5	t@10	t@20
Full (ours)	**48.9**	**68.9**	**82.1**	**21.5**	**40.4**	**59.5**	**29.4**	**52.9**	**70.6**	**3.5**	**9.0**	**18.4**
w/o $\{\boldsymbol{g}_m\}$	43.0	61.7	76.0	21.3	39.6	58.0	11.5	25.1	44.2	2.1	5.6	13.0
w/o $\boldsymbol{c}$	17.4	36.2	54.5	2.1	5.0	10.9	17.4	36.2	54.5	2.1	5.0	10.9
w/o score head	48.0	68.1	81.6	21.4	40.4	59.4	26.5	50.4	69.0	3.2	8.7	17.9
w/o translation	40.5	61.1	77.3	21.0	39.7	58.7	16.4	36.1	58.4	2.3	6.7	15.9
w/o rot6d	43.7	65.0	79.7	21.0	40.1	58.9	20.4	43.3	64.0	2.3	7.5	16.9

Quantitative results. Tab. 3 sweeps N to vary the drift entering refinement. At N=200 the baseline has accumulated little drift and refinement is roughly at parity, while at N=500 and especially N=2000 both rotation and translation AUC improve clearly across all four baselines on both datasets, with the gap widening monotonically with N. This converts streaming pose estimation from an unbounded-error regime into a bounded-error one at the cost of a single refinement pass over the last 100 frames.

Qualitative comparisons. Fig. 4 visualizes a representative ScanNet trajectory under our sliding-window refinement protocol (full details in App. C.3). The raw streaming baselines (blue) drift progressively from ground truth (black); after refinement (red) the refined tail snaps back into close agreement across all three baselines (CUT3R, TTT3R, WINT3R), mirroring the tail-100-frame AUC improvements that Tab. 3 reports. Additional sequences are in App. C.3.

Zero-shot outdoor generalization. We evaluate zero-shot multi-view refinement on Oxford Spires [42] and MatrixCity [20] with CUT3R and TTT3R at $N \in \{200, 500, 2000\}$ (Tab. 4). Both the plane predictor and pose refiner are trained only on indoor data and kept frozen. Rotation AUC improves at $N = 500$ and $N = 2000$, while gains at $N = 200$ are less consistent and some translation metrics decline. Additional tests on CrossLoc [51] drone sequences at $N = 200$ and $N = 500$, together with trajectory construction details, are provided in App. C.1.

4.3 Ablation Study

We ablate POSEPLANER along three axes within Tab. 5: (i) what conditioning the refiner needs (*w/o* $\{\boldsymbol{g}_m\}$, $\boldsymbol{c}$), (ii) how the $M+1$ pose hypotheses are aggregated (*w/o score head*, i.e., the softmax-weighted score-head fusion is replaced by a uniform average), and (iii) how the pose residual is parameterized (*w/o translation* drops the translation channel from $\boldsymbol{z}_0$, leaving a rot-only $\boldsymbol{z}_0 \in \mathbb{R}^6$; *w/o rot6d* replaces the rot6d rotation latent with a unit quaternion in $\mathbb{R}^4$).

Conditioning ablation. Removing the geometry tokens $\{\boldsymbol{g}_m\}$ collapses performance to the init level: without per-match evidence the model can only reproduce the prior mean. Removing the pose token $\boldsymbol{c}$ (equivalently, replacing T_{init} with the identity) forces the model to estimate pose from scratch; the SuperGlue and LoFTR columns become identical since the upstream pose no longer enters the model. The model still recovers a reasonable pose from warping residuals alone, but consistently below the accuracy obtained with a real init.

Score-head fusion vs. uniform averaging. The *w/o score head* variant uniform-averages the $M+1$ pose hypotheses instead of fusing them by the score head's softmax weights. Performance drops: outlier hypotheses receive the same weight as well-aligned ones, whereas the score head learns to discount them via plane-warping cost (Eq. (9)).

Pose-residual parameterization matters. The *w/o translation* variant drops the translation channel from the diffusion residual (rot-only $\boldsymbol{z}_0 \in \mathbb{R}^6$); rotation accuracy degrades because the bilinear coupling between R and $\boldsymbol{t}$ in Δd provides a contextual signal that tightens the rotation prediction. The *w/o rot6d* variant replaces the continuous 6-D rotation representation [59] with a unit quaternion in $\mathbb{R}^4$ and reduces rotation accuracy. Quaternions form a double cover of $\mathrm{SO}(3)$: q and $-q$ represent the same rotation. Choosing a unique quaternion for each rotation introduces discontinuities that can produce abrupt changes in Euclidean flow-matching targets. The continuous 6-D representation avoids this ambiguity. Appendix A.2 provides the mathematical explanation.

Training objective. Flow matching with a single Euler step outperforms direct regression and 100-step DDPM in rotation and translation-direction AUC on ScanNet1500 with LoFTR initialization. Appendix C.4 presents the quantitative results and experimental setup.

4.4 Limitations

POSEPLANER relies on reliable planar evidence, so scenes with limited planar structure or inaccurate plane predictions can reduce refinement gains. The plane source is interchangeable: Appendix C.5 demonstrates refinement with MoGe-derived planes [48], suggesting that improved geometry predictors could further benefit our method. For already-accurate initial poses, plane-prediction noise can outweigh the remaining opportunity for correction and cause slight regressions, as observed for ROMA and MAST3R on ScanNet1500 (Tab. 1). Refinement is therefore most useful when the initialization leaves substantial room for correction; it is not guaranteed to improve an already-near-optimal estimate.

5 Conclusion

We presented POSEPLANER, a flow-matching pose refiner that restores correspondence–pose coupling in feed-forward 3-D reconstruction. It combines single-step rotation refinement conditioned on matched planar primitives with geometric translation recovery for two-view inputs. Trained on two-view pairs, the same refiner improves streaming multi-view poses without retraining. Experiments demonstrate gains across correspondence-based and diffusion-based pose estimators and reduced drift in streaming reconstruction. Extending the conditioning to other geometric primitives is a promising direction for future work.

References

[1] Eric Brachmann and Carsten Rother. Learning less is more - 6d camera localization via 3d surface regression. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 4654–4662. Computer Vision Foundation / IEEE Computer Society, 2018. doi: 10.1109/CVPR.2018.00489. URL http://openaccess.thecvf.com/content_cvpr_2018/html/Brachmann_Learning_Less_Is_CVPR_2018_paper.html.

[2] Eric Brachmann and Carsten Rother. Visual camera re-localization from RGB and RGB-D images using DSAC. *IEEE Trans. Pattern Anal. Mach. Intell.*, 44(9):5847–5865, 2022. doi: 10.1109/TPAMI.2021.3070754. URL https://doi.org/10.1109/TPAMI.2021.3070754.

[3] Eric Brachmann, Alexander Krull, Sebastian Nowozin, Jamie Shotton, Frank Michel, Stefan Gumhold, and Carsten Rother. DSAC - differentiable RANSAC for camera localization. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 2492–2500. IEEE Computer Society, 2017. doi: 10.1109/CVPR.2017.267. URL https://doi.org/10.1109/CVPR.2017.267.

[4] Angel X. Chang, Angela Dai, Thomas A. Funkhouser, Maciej Halber, Matthias Nießner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from RGB-D data in indoor environments. In *2017 International Conference on 3D Vision, 3DV 2017, Qingdao, China, October 10-12, 2017*, pages 667–676. IEEE Computer Society, 2017. doi: 10.1109/3DV.2017.00081. URL https://doi.org/10.1109/3DV.2017.00081.

[5] Avishek Chatterjee and Venu Madhav Govindu. Efficient and robust large-scale rotation averaging. In *IEEE International Conference on Computer Vision, ICCV 2013, Sydney, Australia, December 1-8, 2013*, pages 521–528. IEEE Computer Society, 2013. doi: 10.1109/ICCV.2013.70. URL https://doi.org/10.1109/ICCV.2013.70.

[6] Lin-Zhuo Chen, Jian Gao, Yihang Chen, Ka Leong Cheng, Yipengjing Sun, Liangxiao Hu, Nan Xue, Xing Zhu, Yujun Shen, Yao Yao, et al. Geometric context transformer for streaming 3d reconstruction. *arXiv preprint arXiv:2604.14141*, 2026.

[7] Xingyu Chen, Yue Chen, Yuliang Xiu, Andreas Geiger, and Anpei Chen. TTT3R: 3d reconstruction as test-time training. *CoRR*, abs/2509.26645, 2025. doi: 10.48550/ARXIV.2509.26645. URL https://doi.org/10.48550/arXiv.2509.26645.

[8] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas A. Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 2432–2443. IEEE Computer Society, 2017. doi: 10.1109/CVPR.2017.261. URL https://doi.org/10.1109/CVPR.2017.261.

[9] Afshin Dehghan, Gilad Baruch, Zhuoyuan Chen, Yuri Feigin, Peter Fu, Thomas Gebauer, Daniel Kurz, Tal Dimry, Brandon Joffe, Arik Schwartz, and Elad Shulman. Arkitscenes: A diverse real-world dataset for 3d indoor scene understanding using mobile RGB-D data. In Joaquin Vanschoren and Sai-Kit Yeung, editors, *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*, 2021. URL https://datasets-benchmarks-proceedings.neurips.cc/paper/2021/hash/66f041e16a60928b05a7e228a89c3799-Abstract-round1.html.

[10] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL https://openreview.net/forum?id=YicbFdNTTy.

[11] 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, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pages 19790–19800. IEEE, 2024. doi: 10.1109/CVPR52733.2024.01871. URL https://doi.org/10.1109/CVPR52733.2024.01871.

[12] Martin A. Fischler and Robert C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6): 381–395, 1981. doi: 10.1145/358669.358692. URL https://doi.org/10.1145/358669.358692.

[13] Richard I. Hartley, Jochen Trumpf, Yuchao Dai, and Hongdong Li. Rotation averaging. *Int. J. Comput. Vis.*, 103(3):267–305, 2013. doi: 10.1007/S11263-012-0601-0. URL https://doi.org/10.1007/s11263-012-0601-0.

[14] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/4c5bcfec8584af0d967f1ab10179ca4b-Abstract.html.

[15] Mu Hu, Wei Yin, Chi Zhang, Zhipeng Cai, Xiaoxiao Long, Hao Chen, Kaixuan Wang, Gang Yu, Chunhua Shen, and Shaojie Shen. Metric3d v2: A versatile monocular geometric foundation model for zero-shot metric depth and surface normal estimation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 46(12):10579–10596, 2024. doi: 10.1109/TPAMI.2024.3444912. URL https://doi.org/10.1109/TPAMI.2024.3444912.

[16] Linyi Jin, Shengyi Qian, Andrew Owens, and David F. Fouhey. Planar surface reconstruction from sparse views. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pages 12971–12980. IEEE, 2021. doi: 10.1109/ICCV48922.2021.01275. URL https://doi.org/10.1109/ICCV48922.2021.01275.

[17] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139:1–139:14, 2023. doi: 10.1145/3592433. URL https://doi.org/10.1145/3592433.

[18] Yushi Lan, Yihang Luo, Fangzhou Hong, Shangchen Zhou, Honghua Chen, Zhaoyang Lyu, Shuai Yang, Bo Dai, Chen Change Loy, and Xingang Pan. Stream3r: Scalable sequential 3d reconstruction with causal transformer. *CoRR*, abs/2508.10893, 2025. doi: 10.48550/ARXIV.2508.10893. URL https://doi.org/10.48550/arXiv.2508.10893.

[19] Vincent Leroy, Yohann Cabon, and Jérôme Revaud. Grounding image matching in 3d with mast3r. In Ales Leonardis, Elisa Ricci, Stefan Roth, Olga Russakovsky, Torsten Sattler, and Gül Varol, editors, *Computer Vision - ECCV 2024 - 18th European Conference, Milan, Italy, September 29-October 4, 2024, Proceedings, Part LXXII*, Lecture Notes in Computer Science, pages 71–91. Springer, 2024. doi: 10.1007/978-3-031-73220-1_5. URL https://doi.org/10.1007/978-3-031-73220-1_5.

[20] Yixuan Li, Lihan Jiang, Linning Xu, Yuanbo Xiangli, Zhenzhi Wang, Dahua Lin, and Bo Dai. Matrixcity: A large-scale city dataset for city-scale neural rendering and beyond. In *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*, pages 3182–3192. IEEE, 2023. doi: 10.1109/ICCV51070.2023.00297. URL https://doi.org/10.1109/ICCV51070.2023.00297.

[21] Zizun Li, Jianjun Zhou, Yifan Wang, Haoyu Guo, Wenzheng Chang, Yang Zhou, Haoyi Zhu, Junyi Chen, Chunhua Shen, and Tong He. Wint3r: Window-based streaming reconstruction with camera token pool. *CoRR*, abs/2509.05296, 2025. doi: 10.48550/ARXIV.2509.05296. URL https://doi.org/10.48550/arXiv.2509.05296.

[22] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/forum?id=PqvMRDCJT9t.

[23] Changkun Liu, Bin Tan, Zeran Ke, Shangzhan Zhang, Jiachen Liu, Ming Qian, Nan Xue, Yujun Shen, and Tristan Braud. PLANA3R: zero-shot metric planar 3d reconstruction via feed-forward planar splatting. *CoRR*, abs/2510.18714, 2025. doi: 10.48550/ARXIV.2510.18714. URL https://doi.org/10.48550/arXiv.2510.18714.

[24] Chen Liu, Jimei Yang, Duygu Ceylan, Ersin Yumer, and Yasutaka Furukawa. Planenet: Piecewise planar reconstruction from a single RGB image. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 2579–2588. Computer Vision Foundation / IEEE Computer Society, 2018. doi: 10.1109/CVPR.2018.00273. URL http://openaccess.thecvf.com/content_cvpr_2018/html/Liu_PlaneNet_Piece-Wise_Planar_CVPR_2018_paper.html.

[25] Chen Liu, Kihwan Kim, Jinwei Gu, Yasutaka Furukawa, and Jan Kautz. Planercnn: 3d plane detection and reconstruction from a single image. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 4450–4459. Computer Vision Foundation / IEEE, 2019. doi: 10.1109/CVPR.2019.00458. URL http://openaccess.thecvf.com/content_CVPR_2019/html/Liu_PlaneRCNN_3D_Plane_Detection_and_Reconstruction_From_a_Single_Image_CVPR_2019_paper.html.

[26] Jiachen Liu, Rui Yu, Sili Chen, Sharon X. Huang, and Hengkai Guo. Towards in-the-wild 3d plane reconstruction from a single image. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2025, Nashville, TN, USA, June 11-15, 2025*, pages 27027–27037. Computer Vision Foundation / IEEE, 2025. doi: 10.1109/CVPR52734.2025.02517. URL https://openaccess.thecvf.com/content/CVPR2025/html/Liu_Towards_In-the-wild_3D_Plane_Reconstruction_from_a_Single_Image_CVPR_2025_paper.html.

[27] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/forum?id=XVjTT1nw5z.

[28] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

[29] 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. *Commun. ACM*, 65(1):99–106, 2022. doi: 10.1145/3503250. URL https://doi.org/10.1145/3503250.

[30] Raul Mur-Artal, J. M. M. Montiel, and Juan D. Tardós. ORB-SLAM: A versatile and accurate monocular SLAM system. *IEEE Trans. Robotics*, 31(5):1147–1163, 2015. doi: 10.1109/TRO.2015.2463671. URL https://doi.org/10.1109/TRO.2015.2463671.

[31] David Nistér. An efficient solution to the five-point relative pose problem. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(6):756–777, 2004. doi: 10.1109/TPAMI.2004.17. URL https://doi.org/10.1109/TPAMI.2004.17.

[32] Linfei Pan, Dániel Baráth, Marc Pollefeys, and Johannes L. Schönberger. Global structure-from-motion revisited. In Ales Leonardis, Elisa Ricci, Stefan Roth, Olga Russakovsky, Torsten Sattler, and Gül Varol, editors, *Computer Vision - ECCV 2024 - 18th European Conference, Milan, Italy, September 29-October 4, 2024, Proceedings, Part XL*, Lecture Notes in Computer Science, pages 58–77. Springer, 2024. doi: 10.1007/978-3-031-73661-2_4. URL https://doi.org/10.1007/978-3-031-73661-2_4.

[33] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*, pages 4172–4182. IEEE, 2023. doi: 10.1109/ICCV51070.2023.00387. URL https://doi.org/10.1109/ICCV51070.2023.00387.

[34] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 4937–4946. Computer Vision Foundation / IEEE, 2020. doi: 10.1109/CVPR42600.2020.00499. URL https://openaccess.thecvf.com/content_CVPR_2020/html/Sarlin_SuperGlue_Learning_Feature_Matching_With_Graph_Neural_Networks_CVPR_2020_paper.html.

[35] Manolis Savva, Jitendra Malik, Devi Parikh, Dhruv Batra, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, and Vladlen Koltun. Habitat: A platform for embodied AI research. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 9338–9346. IEEE, 2019. doi: 10.1109/ICCV.2019.00943. URL https://doi.org/10.1109/ICCV.2019.00943.

[36] Johannes L. Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 4104–4113. IEEE Computer Society, 2016. doi: 10.1109/CVPR.2016.445. URL https://doi.org/10.1109/CVPR.2016.445.

[37] Jingjia Shi, Shuaifeng Zhi, and Kai Xu. Planerectr: Unified query learning for 3d plane recovery from a single view. In *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*, pages 9343–9352. IEEE, 2023. doi: 10.1109/ICCV51070.2023.00860. URL https://doi.org/10.1109/ICCV51070.2023.00860.

[38] Jamie Shotton, Ben Glocker, Christopher Zach, Shahram Izadi, Antonio Criminisi, and Andrew W. Fitzgibbon. Scene coordinate regression forests for camera relocalization in RGB-D images. In *2013 IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, June 23-28, 2013*, pages 2930–2937. IEEE Computer Society, 2013. doi: 10.1109/CVPR.2013.377. URL https://doi.org/10.1109/CVPR.2013.377.

[39] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. Loftr: Detector-free local feature matching with transformers. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 8922–8931. Computer Vision Foundation / IEEE, 2021. doi: 10.1109/CVPR46437.2021.00881. URL https://openaccess.thecvf.com/content/CVPR2021/html/Sun_LoFTR_Detector-Free_Local_Feature_Matching_With_Transformers_CVPR_2021_paper.html.

[40] Bin Tan, Nan Xue, Tianfu Wu, and Gui-Song Xia. NOPE-SAC: neural one-plane RANSAC for sparse-view planar 3d reconstruction. *IEEE Trans. Pattern Anal. Mach. Intell.*, 45(12): 15233–15248, 2023. doi: 10.1109/TPAMI.2023.3314745. URL https://doi.org/10.1109/TPAMI.2023.3314745.

[41] Bin Tan, Rui Yu, Yujun Shen, and Nan Xue. Planarsplatting: Accurate planar surface reconstruction in 3 minutes. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2025, Nashville, TN, USA, June 11-15, 2025*, pages 1190–1199. Computer Vision Foundation / IEEE, 2025. doi: 10.1109/CVPR52734.2025.00119. URL https://openaccess.thecvf.com/content/CVPR2025/html/Tan_PlanarSplatting_Accurate_Planar_Surface_Reconstruction_in_3_Minutes_CVPR_2025_paper.html.

[42] Yifu Tao, Miguel Ángel Muñoz-Bañón, Lintong Zhang, Jiahao Wang, Lanke Frank Tarimo Fu, and Maurice Fallon. The oxford spires dataset: Benchmarking large-scale lidar-visual localisation, reconstruction and radiance field methods. *Int. J. Robotics Res.*, 45(6):839–857, 2026. doi: 10.1177/02783649251369905. URL https://doi.org/10.1177/02783649251369905.

[43] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

[44] Hengyi Wang and Lourdes Agapito. 3d reconstruction with spatial memory. In *International Conference on 3D Vision, 3DV 2025, Singapore, March 25-28, 2025*, pages 78–89. IEEE, 2025. doi: 10.1109/3DV66043.2025.00013. URL https://doi.org/10.1109/3DV66043.2025.00013.

[45] Jianyuan Wang, Christian Rupprecht, and David Novotný. Posediffusion: Solving pose estimation via diffusion-aided bundle adjustment. In *IEEE/CVF International Conference on*

Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023, pages 9739–9749. IEEE, 2023. doi: 10.1109/ICCV51070.2023.00896. URL https://doi.org/10.1109/ICCV51070.2023.00896.

[46] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotný. VGGT: visual geometry grounded transformer. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2025, Nashville, TN, USA, June 11-15, 2025*, pages 5294–5306. Computer Vision Foundation / IEEE, 2025. doi: 10.1109/CVPR52734.2025.00499. URL https://openaccess.thecvf.com/content/CVPR2025/html/Wang_VGGT_Visual_Geometry_Grounded_Transformer_CVPR_2025_paper.html.

[47] Qianqian Wang, Yifei Zhang, Aleksander Holynski, Alexei A. Efros, and Angjoo Kanazawa. Continuous 3d perception model with persistent state. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2025, Nashville, TN, USA, June 11-15, 2025*, pages 10510–10522. Computer Vision Foundation / IEEE, 2025. doi: 10.1109/CVPR52734.2025.00983. URL https://openaccess.thecvf.com/content/CVPR2025/html/Wang_Continuous_3D_Perception_Model_with_Persistent_State_CVPR_2025_paper.html.

[48] Ruicheng Wang, Sicheng Xu, Cassie Dai, Jianfeng Xiang, Yu Deng, Xin Tong, and Jiaolong Yang. Moge: Unlocking accurate monocular geometry estimation for open-domain images with optimal training supervision. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2025, Nashville, TN, USA, June 11-15, 2025*, pages 5261–5271. Computer Vision Foundation / IEEE, 2025. doi: 10.1109/CVPR52734.2025.00496. URL https://openaccess.thecvf.com/content/CVPR2025/html/Wang_MoGe_Unlocking_Accurate_Monocular_Geometry_Estimation_for_Open-Domain_Images_with_CVPR_2025_paper.html.

[49] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jérôme Revaud. Dust3r: Geometric 3d vision made easy. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pages 20697–20709. IEEE, 2024. doi: 10.1109/CVPR52733.2024.01956. URL https://doi.org/10.1109/CVPR52733.2024.01956.

[50] Yiming Xie, Matheus Gadelha, Fengting Yang, Xiaowei Zhou, and Huaizu Jiang. Planarrecon: Real-time 3d plane detection and reconstruction from posed monocular videos. *CoRR*, abs/2206.07710, 2022. doi: 10.48550/ARXIV.2206.07710. URL https://doi.org/10.48550/arXiv.2206.07710.

[51] Qi Yan, Jianhao Zheng, Simon Reding, Shanci Li, and Iordan Doytchinov. Crossloc: Scalable aerial localization assisted by multimodal synthetic data. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 17337–17347. IEEE, 2022. doi: 10.1109/CVPR52688.2022.01684. URL https://doi.org/10.1109/CVPR52688.2022.01684.

[52] Yao Yao, Zixin Luo, Shiwei Li, Tian Fang, and Long Quan. Mvsnet: Depth inference for unstructured multi-view stereo. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part VIII*, Lecture Notes in Computer Science, pages 785–801. Springer, 2018. doi: 10.1007/978-3-030-01237-3_47. URL https://doi.org/10.1007/978-3-030-01237-3_47.

[53] Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Nießner, and Angela Dai. Scannet++: A high-fidelity dataset of 3d indoor scenes. In *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*, pages 12–22. IEEE, 2023. doi: 10.1109/ICCV51070.2023.00008. URL https://doi.org/10.1109/ICCV51070.2023.00008.

[54] Zehao Yu, Jia Zheng, Dongze Lian, Zihan Zhou, and Shenghua Gao. Single-image piece-wise planar 3d reconstruction via associative embedding. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 1029–1037. Computer Vision Foundation / IEEE, 2019. doi: 10.1109/CVPR.2019.00112. URL http://openaccess.thecvf.com/content_CVPR_2019/html/Yu_Single-Image_Piece-Wise_Planar_3D_Reconstruction_via_Associative_Embedding_CVPR_2019_paper.html.

[55] Shuai Yuan, Yantai Yang, Xiaotian Yang, Xupeng Zhang, Zhonghao Zhao, Lingming Zhang, and Zhipeng Zhang. Infinitevggt: Visual geometry grounded transformer for endless streams. *CoRR*, abs/2601.02281, 2026. doi: 10.48550/ARXIV.2601.02281. URL https://doi.org/10.48550/arXiv.2601.02281.

[56] Jason Y. Zhang, Amy Lin, Moneish Kumar, Tzu-Hsuan Yang, Deva Ramanan, and Shubham Tulsiani. Cameras as rays: Pose estimation via ray diffusion. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=EanCFCwAjM.

[57] 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, CVPR 2025, Nashville, TN, USA, June 11-15, 2025*, pages 6317–6326. Computer Vision Foundation / IEEE, 2025. doi: 10.1109/CVPR52734.2025.00592. URL https://openaccess.thecvf.com/content/CVPR2025/html/Zhao_DiffusionSfM_Predicting_Structure_and_Motion_via_Ray_Origin_and_Endpoint_CVPR_2025_paper.html.

[58] Wang Zhao, Jiachen Liu, Sheng Zhang, Yishu Li, Sili Chen, Sharon X. Huang, Yong-Jin Liu, and Hengkai Guo. Monoplane: Exploiting monocular geometric cues for generalizable 3d plane reconstruction. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2024, Abu Dhabi, United Arab Emirates, October 14-18, 2024*, pages 8481–8488. IEEE, 2024. doi: 10.1109/IROS58592.2024.10802672. URL https://doi.org/10.1109/IROS58592.2024.10802672.

[59] Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 5745–5753. Computer Vision Foundation / IEEE, 2019. doi: 10.1109/CVPR.2019.00589. URL http://openaccess.thecvf.com/content_CVPR_2019/html/Zhou_On_the_Continuity_of_Rotation_Representations_in_Neural_Networks_CVPR_2019_paper.html.

[60] Dong Zhuo, Wenzhao Zheng, Jiahe Guo, Yuqi Wu, Jie Zhou, and Jiwen Lu. Streaming 4d visual geometry transformer. *CoRR*, abs/2507.11539, 2025. doi: 10.48550/ARXIV.2507.11539. URL https://doi.org/10.48550/arXiv.2507.11539.

Technical Appendices and Supplementary Material

A Geometric and Algorithmic Foundations

This section formalizes three pieces of background that the main text builds on: the plane-warping residuals used as conditioning (Sec. A.1), the continuous 6-D rotation representation carried by the pose latent (Sec. A.2), and the flow-matching parameterization under which a single Euler step at inference is sufficient (Sec. A.3).

A.1 Plane-Warping Residuals as Conditioning Signal

Vanishing at the true pose; monotonic away from it. Let $\boldsymbol{e}_n(R) := \boldsymbol{n}_1 - R\boldsymbol{n}_2$ and $e_d(R, \boldsymbol{t}) := d_1 - d_2 + (R\boldsymbol{n}_2)^\top \boldsymbol{t}$ denote the (vector / signed) raw residuals whose norm and absolute value are the $\Delta\boldsymbol{n}$ and Δd of Eq. (2). By construction, $\boldsymbol{e}_n(R_{\text{gt}}) = \mathbf{0}$ and $e_d(R_{\text{gt}}, \boldsymbol{t}_{\text{gt}}) = 0$ identically. Under a joint first-order perturbation $R = \exp([\delta\boldsymbol{r}]_\times) R_{\text{gt}}$ and $\boldsymbol{t} = \boldsymbol{t}_{\text{gt}} + \delta\boldsymbol{t}$, these raw residuals expand to

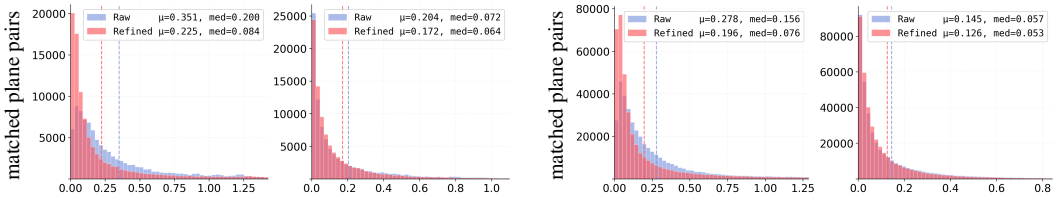
$$\boldsymbol{e}_n(R) \;\approx\; -[\delta\boldsymbol{r}]_\times \, R_{\text{gt}}\boldsymbol{n}_2 \,, \tag{14}$$

$$e_d(R, \boldsymbol{t}) \;\approx\; \big(R_{\text{gt}}\boldsymbol{n}_2\big)^\top \delta\boldsymbol{t} \;+\; \big([\delta\boldsymbol{r}]_\times R_{\text{gt}}\boldsymbol{n}_2\big)^\top \boldsymbol{t}_{\text{gt}} \,. \tag{15}$$

$\Delta\boldsymbol{n} = \|\boldsymbol{e}_n\|_2$ is therefore linear in the rotation error along the angle between $\delta\boldsymbol{r}$ and $R_{\text{gt}}\boldsymbol{n}_2$; $\Delta d = |e_d|$ contains a translation-only probe along the direction $R_{\text{gt}}\boldsymbol{n}_2$ *plus* a rotation-coupled term whose strength scales with $\|\boldsymbol{t}_{\text{gt}}\|$. A set of M correspondences with diverse normals therefore exposes the full $\delta\boldsymbol{r}/\delta\boldsymbol{t}$ error along M different directions, effectively a dense per-direction error map.

Estimator-agnostic by construction. Eqs. (14)–(15) read the residuals off paired plane primitives and a candidate pose alone, with no dependency on how that pose was produced. The same conditioning signal is therefore presented to the refiner regardless of upstream—feed-forward network, matcher-plus-solver, or identity initialization.

Empirical verification: the refiner implicitly learns the warp. In contrast to NOPESAC [40], which explicitly warps view-1's plane parameters by the initial pose to feed the network a pre-aligned representation, POSEPLANER provides only the raw per-view primitives $(\boldsymbol{n}_m^1, \bar{d}_m^1, \boldsymbol{n}_m^2, \bar{d}_m^2)$ in each camera's own local frame; the warping residuals $(\Delta\boldsymbol{n}, \Delta d)$ enter only as detached geometric weights ω_m (Eq. (8)) and the auxiliary $\mathcal{L}_{\text{geo}}$ (Eq. (11)), never as inputs. This setup encourages the network to learn the warping algebra from the geometric supervision rather than rely on input-space normalization. Fig. 5 suggests the network does pick up this implicit warp: across all matched plane pairs, both histograms collapse markedly toward zero after refinement—median $\|\Delta\boldsymbol{n}\|_2$ drops from 0.200 to 0.084 and median $|\Delta d|$ from 0.072 to 0.064—even though neither residual is fed in, and the effect transfers across ScanNet1500 (in-distribution) and 7Scenes (zero-shot).



(a) ScanNet1500, upstream = LOFTR (b) 7Scenes, upstream = LOFTR

Figure 5: **Refinement collapses the plane-warping residuals.** (a) ScanNet1500 and (b) 7Scenes, LOFTR upstream. Each sub-figure: $\|\Delta\boldsymbol{n}\|_2$ (left) and $|\Delta d|$ (right, scene-normalized) over all matched plane pairs; **blue** = raw, **red** = after POSEPLANER; dashed lines mark means.

A.2 Rotation as a Continuous 6-D Representation

The rotation part of the pose residual $\boldsymbol{z}_0$ in Eq. (6) is encoded in the 6-D representation of Zhou et al. [59], not as a unit quaternion. Under a flow-matching objective this choice is forced, not cosmetic: quaternions carry a topological discontinuity that interacts destructively with continuous probability paths.

Formal definition of a continuous representation. Following Zhou et al. [59], a *representation* of $\mathrm{SO}(3)$ with domain $X \subseteq \mathbb{R}^d$ is a pair of maps

$$f : \mathrm{SO}(3) \to X, \qquad g : X \to \mathrm{SO}(3), \qquad g \circ f = \mathrm{id}_{\mathrm{SO}(3)}, \tag{16}$$

and the representation is *continuous* when both f and g are continuous under the Euclidean topology of $\mathbb{R}^d$. Neural networks are continuous under this topology, so a discontinuous g forces the network to reproduce a discontinuity somewhere inside its training support.

The $d \leq 4$ obstruction. Zhou et al. [59] prove that no continuous representation of $\mathrm{SO}(3)$ exists for $d \leq 4$. Concretely, the discontinuity always lives in the encoder f, since the polynomial decoders typically used for these representations are themselves continuous:

- **Euler angles** (d=3) have a gimbal-lock singularity where f is not unique and g is not one-to-one.
- **Axis-angle** (d=3) maps $\mathrm{SO}(3)$ to a closed ball of radius π, with antipodal points on the boundary $\|\boldsymbol{r}\| = \pi$ identified to the same rotation; the encoder f must therefore choose one of the two boundary representatives, creating a sign-flip discontinuity at $\|\boldsymbol{r}\| = \pi$.
- **Unit quaternions** (d=4) form a *double cover* of $\mathrm{SO}(3)$: every R maps to an antipodal pair $\pm\boldsymbol{q}$ on S^3. The decoder $g : \mathbb{R}^4 \to \mathrm{SO}(3)$ is the standard quaternion-to-matrix polynomial, which is continuous everywhere; the obstruction lies in the encoder $f : \mathrm{SO}(3) \to S^3$, which must select one of $\pm\boldsymbol{q}$ globally and therefore admits no continuous global section—arbitrarily close rotations either side of the chosen branch are encoded to far-apart quaternions of opposite sign.

$\mathrm{SO}(3)$ is compact and topologically non-trivial, so no homeomorphism to $\mathbb{R}^d$ exists for any finite d; a continuous *embedding* is the best one can hope for, and Zhou et al. [59] show $d = 5$ is the minimum ($d = 6$ is the simplest practical construction).

The 6-D continuous representation. The encoder drops the third column of R,

$$f\big([\,\boldsymbol{r}_1 \mid \boldsymbol{r}_2 \mid \boldsymbol{r}_3\,]\big) \;=\; [\boldsymbol{r}_1;\, \boldsymbol{r}_2] \in \mathbb{R}^6. \tag{17}$$

The decoder $g : \mathbb{R}^6 \to \mathrm{SO}(3)$ is Gram–Schmidt orthonormalization on the two column vectors: given $(\boldsymbol{a}_1, \boldsymbol{a}_2) \in \mathbb{R}^3 \times \mathbb{R}^3$ linearly independent,

$$\boldsymbol{b}_1 = N(\boldsymbol{a}_1), \quad \boldsymbol{b}_2 = N\big(\boldsymbol{a}_2 - (\boldsymbol{b}_1 \cdot \boldsymbol{a}_2)\,\boldsymbol{b}_1\big), \quad \boldsymbol{b}_3 = \boldsymbol{b}_1 \times \boldsymbol{b}_2, \tag{18}$$

with $g(\boldsymbol{a}_1, \boldsymbol{a}_2) = [\boldsymbol{b}_1 \mid \boldsymbol{b}_2 \mid \boldsymbol{b}_3]$ and $N(\boldsymbol{v}) = \boldsymbol{v}/\|\boldsymbol{v}\|_2$. Both f and g are continuous on the full-measure open subset of $\mathbb{R}^6$ where $\boldsymbol{a}_1, \boldsymbol{a}_2$ are linearly independent, and $g \circ f = \mathrm{id}_{\mathrm{SO}(3)}$ by construction.

Why continuity is critical under flow matching. The flow-matching path of Eq. (19) is a straight line in the ambient Euclidean space, not on S^3. The encoder f enters only at the endpoint $\boldsymbol{z}_0 = f(R_{\text{gt}} R_{\text{init}}^{\top})$, but precisely because f is discontinuous along the antipodal branch cut, two arbitrarily close ground-truth rotations on either side of the cut are encoded to quaternions of opposite sign $\pm\boldsymbol{q}$, producing two FM target velocities $\boldsymbol{v}^\star = \boldsymbol{\epsilon} - (\pm\boldsymbol{q})$ that disagree by $2\boldsymbol{q}$ for the *same* input geometric condition. The network is then forced to average two conflicting targets near the cut, biasing the learned velocity toward the conditional mean of the bimodal label distribution rather than either branch. With rot6d, the encoder f is continuous on the open dense subset where R has linearly independent first two columns, so $\boldsymbol{z}_0$ varies continuously with R and $\boldsymbol{v}^\star$ is single-valued at every $(\boldsymbol{z}_t, t)$. The one-step Euler guarantee established in App. A.3 therefore holds end-to-end only when the encoder is continuous.

A.3 Flow Matching as a Linear Denoising Process

This subsection reviews the flow-matching objective used by the refiner and then explains why a *single* Euler step at inference is sufficient, via an analogy to Newton's method on linear systems.

Flow-matching preliminaries. Flow matching [22, 27] defines a continuous probability path $p_t(\boldsymbol{z})$ between a tractable prior $p_1 = \mathcal{N}(0, \boldsymbol{I})$ and the data distribution p_0 through a *straight-line* interpolation,

$$\boldsymbol{z}_t \;=\; (1 - t)\,\boldsymbol{z}_0 + t\,\boldsymbol{\epsilon}, \qquad \boldsymbol{\epsilon} \sim \mathcal{N}(0, \boldsymbol{I}), \qquad t \in [0, 1]. \tag{19}$$

Differentiating with respect to t gives the instantaneous target velocity

$$\boldsymbol{v}^\star(\boldsymbol{z}_t, t) \;=\; \frac{d\boldsymbol{z}_t}{dt} \;=\; \boldsymbol{\epsilon} - \boldsymbol{z}_0, \tag{20}$$

which is *constant along the line* connecting $\boldsymbol{z}_0$ and $\boldsymbol{\epsilon}$. The training objective regresses a learned velocity field $\boldsymbol{v}_\theta(\boldsymbol{z}_t, t)$ to this target,

$$\mathcal{L}_{\mathrm{FM}} \;=\; \mathbb{E}_{t,\boldsymbol{z}_0,\boldsymbol{\epsilon}}\big[\big\|\boldsymbol{v}_\theta(\boldsymbol{z}_t, t) - \boldsymbol{v}^\star\big\|_1\big]\,. \tag{21}$$

At inference, $\boldsymbol{z}_0$ is recovered by integrating the learned ODE from $t = 1$ to $t = 0$,

$$\boldsymbol{z}_0 \;=\; \boldsymbol{z}_1 - \int_0^1 \boldsymbol{v}_\theta(\boldsymbol{z}_t, t)\, dt. \tag{22}$$

Comparison with DDPM. Standard DDPM [14] regresses either $\boldsymbol{\epsilon}$ or $\boldsymbol{z}_0$ under a variance-preserving schedule whose probability path is curved (a geodesic on a Gaussian manifold, not a straight line). Integrating a curved path requires many steps to stay close to it, so DDPM samplers typically use 50–1000 steps. Flow matching replaces the curve with a line, so the integrand in Eq. (22) becomes constant in t and sampling accuracy decouples from step count.

Why one Euler step suffices: conditional-mean regression at $t = 1$. The most striking empirical property of our refiner is that a *single* Euler step at inference (Eq. (12)) matches a many-step DDPM sampler. The clean justification is not that the learned ODE is constant along paths—it is not, since the network regresses the *marginal* velocity $\boldsymbol{v}_\theta(\boldsymbol{z}_t, t) = \mathbb{E}[\boldsymbol{\epsilon} - \boldsymbol{z}_0 \mid \boldsymbol{z}_t, t]$ rather than the sample-specific $\boldsymbol{v}^\star$—but rather that, at the single time $t = 1$ we use, this marginal velocity reduces to a conditional-mean estimator of $\boldsymbol{z}_0$. Concretely, since $\boldsymbol{z}_1 = \boldsymbol{\epsilon} \sim \mathcal{N}(0, \boldsymbol{I})$ is independent of $\boldsymbol{z}_0$ under the FM coupling,

$$\boldsymbol{v}_\theta(\boldsymbol{\epsilon}, 1) \;=\; \mathbb{E}\big[\boldsymbol{\epsilon} - \boldsymbol{z}_0 \,\big|\, \boldsymbol{z}_1 = \boldsymbol{\epsilon}\big] \;=\; \boldsymbol{\epsilon} - \mathbb{E}[\boldsymbol{z}_0 \mid c], \tag{23}$$

where c collects the conditioning tokens (geometry $\{\hat{\boldsymbol{g}}_m\}$ and pose $\hat{\boldsymbol{c}}$). A single forward Euler step from $\boldsymbol{z}_1$ therefore yields

$$\boldsymbol{z}_1 - \boldsymbol{v}_\theta(\boldsymbol{z}_1, 1) \;=\; \boldsymbol{\epsilon} - \big(\boldsymbol{\epsilon} - \mathbb{E}[\boldsymbol{z}_0 \mid c]\big) \;=\; \mathbb{E}[\boldsymbol{z}_0 \mid c]. \tag{24}$$

That is, single-step FM inference is mathematically equivalent to direct conditional-mean regression of $\boldsymbol{z}_0$ given the geometric conditioning. Additional Euler steps between $t=1$ and $t=0$ only matter when the marginal field is non-affine in t along the integration trajectory; they cannot improve the $t=1$ readout, which is already an unbiased estimator of $\mathbb{E}[\boldsymbol{z}_0 \mid c]$.

Newton-style intuition. The structural reason this conditional-mean readout is sharp in our setting is the same one that makes Newton's method exact on linear systems: a tractable forward map admits a closed-form inverse, so iteration is only useful when the map is non-affine. Here the FM forward process $\boldsymbol{z}_t = (1 - t)\boldsymbol{z}_0 + t\boldsymbol{\epsilon}$ is affine in t, and the conditioning c collapses the residual distribution $p(\boldsymbol{z}_0 \mid c)$ to a low-variance, near-unimodal cluster (the pose-residual is small once a sensible T_{init} has been provided). Three properties of our problem keep this conditional-mean estimator low-variance:

- The latent is 9-dimensional, far lower than image-scale diffusion, so the marginal velocity is an easy function to fit.
- The pose-residual is concentrated near a fixed point in this 9-D space by construction, via scene normalization and the residual parameterization of Eq. (6), so $p(\boldsymbol{z}_0 \mid c)$ is sharply peaked.
- The FM path is continuous end to end (App. A.2), so the conditional mean varies smoothly with c.

A single Euler step at $t=1$ is therefore the natural inference choice for pose-residual denoising under our flow-matching probability path.

B Model and Training Details

B.1 Plane Head Training Details

This section gives the plane-head representation, the per-view local-frame convention, and the training objective omitted from Sec. 3.1.

Table 6: **Training data and sample sizes.** Same mixture used by PLANA3R [23].

Dataset	Type	# Pairs
SCANNETV2 [8]	Indoor / Real	610K
SCANNET++ [53]	Indoor / Real	810K
ARKITSCENES [9]	Indoor / Real	2,400K
HABITAT [35]	Indoor / Synthetic	120K
Total		**$\sim$3,940K**

Representation. The plane head produces a *patch-level* planar description of each view: every decoder token emits one plane $(\boldsymbol{n}, d)$, giving one normal-offset pair per patch. Following PLANA3R [23], each token regresses not $(\boldsymbol{n}, d)$ directly but a compact tuple $(d_\pi^i, \boldsymbol{r}_\pi^i, \boldsymbol{q}_\pi^{i,j})$: the center depth along the token's viewing ray in source view I^i, two elliptical radii defining the primitive's disk footprint, and a unit quaternion expressed in the frame of a reference camera I^j. From this tuple,

$$\boldsymbol{n}_\pi \;=\; R(\boldsymbol{q}_\pi^{i,j})\,[0,0,1]^\top, \qquad d_\pi \;=\; -\boldsymbol{n}_\pi^\top \boldsymbol{P}_\pi, \tag{25}$$

with $\boldsymbol{P}_\pi$ the primitive's 3-D center in I^j, so $\boldsymbol{n}_\pi$ and d_π live in I^j. Our sole departure from PLANA3R is the choice of j: the original fixes $j{=}1$ (a single global frame, which forces view-2 primitives to be transformed through the predicted pose before supervision), whereas we set $j{=}i$, so every primitive lives in its *own source camera's frame*. Three consequences follow. **(i)** Plane prediction decouples from pose and scales directly to $n \geq 2$ views. **(ii)** The plane-head loss decomposes per view: each view's primitives are compared only against its own ground truth, so no pose enters the plane-head gradient. Training still uses image pairs, and we keep PLANA3R's paired sampling, siamese encoder, and cross-attention decoders to preserve metric-scale consistency within each pair. **(iii)** The warping identity (1) becomes *observable*: the two views produce genuinely distinct $(\boldsymbol{n}^1, d^1)$ and $(\boldsymbol{n}^2, d^2)$, so the residuals $(\Delta\boldsymbol{n}, \Delta d)$ of Eq. (2) turn into non-trivial conditioning signals that carry the pose error into Sec. 3.2. All remaining components (ViT encoder, two cross-attention decoders, the hierarchical primitive prediction architecture (HPPA) with low $\frac{H}{16}\times\frac{W}{16}$ and high $\frac{H}{8}\times\frac{W}{8}$ resolutions selectively fused, and the PLANARSPLATTING rasterizer [41]) are inherited from PLANA3R unchanged.

Training objective and refiner input. We supervise the head with the *plane-only* losses of PLANA3R, applied per view in its own local frame. Following [23, Eq. 3], the patch loss on each HPPA resolution $* \in \{\text{low}, \text{high}\}$ is

$$\mathcal{L}_*^{\text{patch}} \;=\; \alpha_1 \big\| 1 - (\boldsymbol{N}_*^{\text{patch}})^\top \boldsymbol{N}_*^{\text{r, gt}} \big\|_1 + \alpha_1 \big\| \boldsymbol{N}_*^{\text{patch}} - \boldsymbol{N}_*^{\text{r, gt}} \big\|_1 + \alpha_2 \big\| \boldsymbol{D}_*^{\text{patch}} - \boldsymbol{D}_*^{\text{r, gt}} \big\|_1, \tag{26}$$

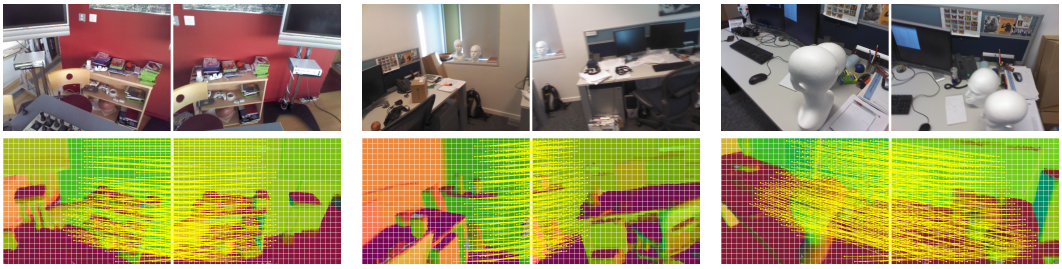
where $(\boldsymbol{N}_*^{\text{r, gt}}, \boldsymbol{D}_*^{\text{r, gt}})$ are the source-view ground-truth normal and depth maps resized to the HPPA grid. The rendering loss $\mathcal{L}_*^{\text{render}}$ [23, Eq. 4] has the identical functional form, substituting $\boldsymbol{N}_*^{\text{patch}} \to \boldsymbol{N}_*^{\text{render}}$ and $\boldsymbol{D}_*^{\text{patch}} \to \boldsymbol{D}_*^{\text{render}}$ (the full-resolution PLANARSPLATTING renderings) against the original-resolution $(\boldsymbol{N}^{\text{gt}}, \boldsymbol{D}^{\text{gt}})$ with weights (β_1, β_2) and index $* \in \{\text{low}, \text{high}, \text{selected}\}$. The plane-head objective sums the two losses over the two views,

$$\mathcal{L}_{\text{plane}} \;=\; \sum_{i=1,2} \big[\mathcal{L}^{\text{patch},(i)} + \mathcal{L}^{\text{render},(i)} \big]. \tag{27}$$

We drop the pose loss of the original PLANA3R [23, Eq. 5]: under our local-frame convention it is both unnecessary (pose is not part of the head's output) and inconsistent with the per-view decomposition above. At inference, the refiner consumes only the low-resolution primitives $\{(\boldsymbol{n}_m^i, d_m^i)\}_{m=1}^M$ from the native $\frac{H}{16}\times\frac{W}{16}$ HPPA grid (without the deconvolution upsampler), yielding M plane correspondences per pair; this keeps M small enough for full self-attention in the refiner and retains the resolution at which the planar assumption is most reliable.

Training data. Both the plane head and the refiner are trained on the same four-indoor-scene mixture used by PLANA3R [23], summarized in Tab. 6: $\sim$3.94M two-view pairs from SCANNETV2 [8], SCANNET++ [53], ARKITSCENES [9], and HABITAT [35]. Pseudo ground-truth normal maps are generated by METRIC3DV2 [15]; depth maps are taken directly from the datasets.

Visualization. Fig. 6 visualizes both the plane head's per-pixel rasterized normal and the MAST3R [19] patch correspondences entering the refiner as $\{\boldsymbol{g}_m\}_{m=1}^M$ (Eq. (3)) on three 7SCENES pairs. Color-uniform tiles correspond to single plane primitives; the yellow corr lines pair patches that, after rasterization, look like the same physical plane viewed from two cameras.



(a) CHESS, `seq-01` (b) OFFICE, `seq-01` (c) STAIRS, `seq-01`

Figure 6: **Predicted plane primitives and matched patch correspondences.** Three 7SCENES pairs (columns). **Top**: input RGB (view 1 | view 2). **Bottom**: PLANARSPLATTING-rasterized normal from our plane head with the 24×32 patch grid (white); yellow lines connect MAST3R's matched patch centers, which form $\{\boldsymbol{g}_m\}_{m=1}^{M}$ in Eq. (3).

B.2 Refiner Architecture Details

This section expands the multi-query denoiser introduced in Sec. 3.2 with the per-query construction, attention depth, output head, and the precomputable-context optimization that the single-step inference relies on.

Per-query latent-time and geometry bias. The latent-time component of Eq. (7) uses $\phi_z : \mathbb{R}^9 \to \mathbb{R}^d$ and $\phi_t : \mathbb{R}^{d_t} \to \mathbb{R}^d$ as small MLPs, with $\mathrm{SinEmb}(\cdot)$ a sinusoidal time embedding. The per-match geometry bias $\boldsymbol{q}_m = \boldsymbol{q}_0 + W_b\,\hat{\boldsymbol{g}}_m$, with learned $W_b \in \mathbb{R}^{d\times d}$, grounds query m in correspondence m.

Cross-attention stack and output head. The M queries pass through $L_{\text{den}}{=}4$ pre-norm cross-attention blocks attending to the context $\hat{\boldsymbol{C}}$, followed by a $\mathrm{LayerNorm}{+}\mathrm{Linear}(d, 9)$ head that emits the per-match velocity predictions $\boldsymbol{v}_m(\boldsymbol{z}_t, t)$.

Precomputable context. The time step t modulates only the queries (no AdaLN-Zero on $\hat{\boldsymbol{C}}$), so $\hat{\boldsymbol{C}}$ is computed once per pair and reused across all queries and Euler steps—a natural fit for the single-step inference of Sec. 3.2.

B.3 Training and Inference Details

Implementation. The plane head adopts the training recipe of PLANA3R [23] adapted to the per-view local-frame convention of Sec. 3.1: the ViT encoder and the two cross-attention decoders are initialized from DUST3R's 512-DPT pre-trained weights, optimized with AdamW [28] (learning rate $1\times10^{-4} \to 1\times10^{-6}$) under the plane-related losses of Eq. (27) at input resolution 512×384, with a one-epoch warm-up under the patch loss alone followed by ten epochs jointly optimizing patch and rendering losses; the gradient threshold for selectively fusing the low- and high-resolution primitives is $g_{\text{th}}{=}0.5$. With the plane head frozen, the diffusion refiner and the score head are then trained on our four-dataset mixture (Tab. 6) with Eq. (11) for 20 epochs at per-GPU batch size 12, using AdamW with the same $1\times10^{-4} \to 1\times10^{-6}$ schedule, $L_{\text{cond}} = L_{\text{den}} = 4$ Transformer blocks, and M query tokens per pair. At inference, a single Euler step (Eq. (12)) yields the refined rotation $R^\star$; translation is recovered by the adaptively regularized Levenberg–Marquardt solver of Eq. (13). Total compute is 128 GPU-days on NVIDIA A800 GPUs.

The remainder of this subsection details the inference-time solvers omitted from Sec. 3.2.

Scene-range normalization. The per-pair scalar σ that non-dimensionalises d in Eq. (5) is the mean absolute plane offset across all S patches per view,

$$\sigma \;=\; \tfrac{1}{2S}\Big(\textstyle\sum_{p=1}^{S} |d_p^1| + \sum_{p=1}^{S} |d_p^2|\Big), \tag{28}$$

which makes $\bar{d}_m^v = d_m^v/\sigma$ approximately $O(1)$ across pairs.

Two-view translation solver (Eq. (13)). Locking $s = \|\boldsymbol{t}_{\text{init}}\|$, the two data residuals are

$$\Delta d_m(\hat{\boldsymbol{t}}) \;=\; d_m^1 - d_m^2 + (R^\star \boldsymbol{n}_m^2)^\top (s\hat{\boldsymbol{t}}), \tag{29}$$

$$\Delta \boldsymbol{P}_m(\hat{\boldsymbol{t}}) \;=\; \boldsymbol{P}_m^1 - R^\star \boldsymbol{P}_m^2 - s\hat{\boldsymbol{t}}, \tag{30}$$

where $\boldsymbol{P}_m^v = z_m^v K_v^{-1}[u_m; 1]$ is the back-projection of matched pixel u_m through the dense depth z_m^v rasterized by PLANARSPLATTING from the per-patch plane primitives. Both ρ_d, ρ_p are Huber losses with threshold 0.1 and weights $w_d = w_p = 1$. The init-anchored Tikhonov weight λ_{reg} adapts to the init's quality: with $r_m^{(0)} = (R^\star \boldsymbol{n}_m^2)^\top \boldsymbol{t}_{\text{init}} - (d_m^2 - d_m^1)$ the Δd residual at $\hat{\boldsymbol{t}}_{\text{init}}$, we set $\sigma_{\text{init}} = 1.4826\,\mathrm{MAD}(r^{(0)})$ on its 20–80% IQR-trimmed core and

$$\lambda_{\text{reg}} = \mathrm{clip}\big(\lambda_0(\sigma_{\text{ref}}/\sigma_{\text{init}})^\alpha,\ \lambda_{\min},\ \lambda_{\max}\big), \quad (\lambda_0, \sigma_{\text{ref}}, \alpha, \lambda_{\min}, \lambda_{\max}) = (200,\ 0.5,\ 4,\ 1,\ 10^6), \tag{31}$$

so a small σ_{init} (init already consistent with the predicted planes) inflates λ_{reg} and the solver returns $\approx \hat{\boldsymbol{t}}_{\text{init}}$, while a large σ_{init} lets the data terms drive the solve. We optimize via `scipy.optimize.least_squares` (TRF, Huber $f_{\text{scale}} = 0.1$) over $\hat{\boldsymbol{t}} \in S^2$, ≤ 15 Jacobian evaluations.

Multi-view aggregation (n-view). For streaming n-view inputs we follow the GLOMAP [32] global-SfM pipeline—rotation averaging, multi-view track construction, point-based global positioning, and bundle adjustment—over the refinement-window edge set $\mathcal{E}$. The diffusion refiner is invoked only for the per-edge *rotation* $R_{ij}^\star$; the per-edge translation $\boldsymbol{t}_{ij}^{\text{raw}}$ is taken directly from the upstream estimator and used unchanged downstream.

Rotation averaging. Per-frame absolute rotations are recovered from the pairwise $\{R_{ij}^\star\}$ by robust least-squares [13, 5],

$$\{R_i\}^\star = \arg\min_{\{R_i\}} \sum_{(i,j)\in\mathcal{E}} \rho\big(d(R_j^\top R_{ij}^\star R_i,\ \boldsymbol{I})\big)\,, \tag{32}$$

with ρ a Huber loss and d chordal distance.

Tracks and translation warm-start. Multi-view tracks $\{\tau_k\}$ are built by union-find over pixel-cell-quantized correspondences from the matcher; the upstream's raw pairwise translations $\{\boldsymbol{t}_{ij}^{\text{raw}}\}$ together with $\{R_i\}^\star$ are composed into an absolute c2w trajectory and used as warm-start camera centers for the positioner below.

Global positioning. With $\{R_i\}$ fixed and tracks $\{\tau_k\}$ attached with bearings $\boldsymbol{v}_{ki} = R_i^\top K_i^{-1}[u_{ki}; 1]$, GLOMAP's point-based mode [32] jointly optimizes camera centers, 3-D point positions, and per-observation scales,

$$\{\boldsymbol{c}_i, \boldsymbol{X}_k, s_{ki}\}^\star = \arg\min \sum_k \sum_{(k,i)\in\tau_k} \rho(\|\boldsymbol{v}_{ki} - s_{ki}(\boldsymbol{X}_k - \boldsymbol{c}_i)\|_2)\,, \quad s_{ki} \geq 0. \tag{33}$$

Bundle adjustment. A Huber-robust BA [32] over $(\boldsymbol{c}_i, \boldsymbol{X}_k)$ with pinhole reprojection error refines centers and 3-D points; we run GLOMAP with `fix_rotations=True` so that $\{R_i\}^\star$ from rotation averaging remain frozen throughout positioning and BA, and the entire downstream pipeline only ever consumes the refined rotations. The two-view LM translation solver of Eq. (13) is bypassed entirely at n-view scale, since the dense edge set $\mathcal{E}$ already over-constrains every $\boldsymbol{c}_i$ through the RA + GP + BA pipeline; the same plane-conditioned refiner—whose only role at n-view is to produce $\{R_{ij}^\star\}$ per edge—therefore serves both regimes without retraining.

C Additional Evaluation Setup and Results

C.1 Pair and Sequence Construction

This subsection gives the per-sequence quantitative breakdown deferred from Sec. 4 for the two-view and streaming evaluation splits.

7Scenes two-view pair list. We use all 46 sequences across the seven scenes (chess, fire, heads, office, pumpkin, redkitchen, stairs) without any train/test restriction. Pairs are formed as $(i,\ i{+}50)$ at stride 10 within each sequence, producing 95 pairs per 1000-frame sequence and 45 per 500-frame *stairs* sequence, totalling $40\times95 + 6\times45 = 4070$ pairs. The exact pair list is released with our code.

Matterport3D two-view pair list. For an additional zero-shot test, we use the 7996-pair test set of SPARSEPLANES [16] on Matterport3D [4]. Results in Tab. 7 mirror the pattern from ScanNet1500 and 7Scenes: POSEPLANER delivers a large gain on the weaker LOFTR upstream (R@5: $10.7 \rightarrow 20.2$,

Table 7: **Two-view rotation refinement on Matterport3D.** Evaluated zero-shot on the 7996-pair test set of SPARSEPLANES [16]. Rotation AUC@τ° for two upstream matchers in two configurations: *init* and +POSEPLANER. The *Valid* column counts pairs with a valid upstream pose; AUC is computed only on this subset. Higher is better; best per block in **bold**.

Method	LOFTR [39]				SUPERGLUE [34]			
	R@5	R@10	R@20	Valid	R@5	R@10	R@20	Valid
init	10.7	18.8	30.6	4385/7996	38.3	53.5	66.7	6853/7996
+ POSEPLANER	**20.2**	**36.0**	**52.8**		**40.8**	**57.2**	**70.6**	

R@20: $30.6 \rightarrow 52.8$) and a smaller but consistent gain on the stronger SUPERGLUE upstream (R@5: $38.3 \rightarrow 40.8$, R@20: $66.7 \rightarrow 70.6$).

ScanNet streaming. The 100 trajectories of the ScanNet1500 test set (scenes 0707–0806) range in raw length from 163 to 4168 frames; we sample at frame gap 2 to obtain the streaming sequences. For each input-length threshold $N \in \{200, 500, 2000\}$ we retain trajectories with $\geq 2N$ raw frames (equivalently $\geq N$ sampled frames)—99, 86, and 7 respectively—and let the streaming baseline ingest the first N sampled frames while POSEPLANER refines the last 100.

7Scenes streaming. We form N-frame trajectories at frame gap 2 from the same 46 sequences as the two-view setup. For $N \in \{200, 500\}$ we take the first N sampled frames of every sequence with $\geq 2N$ raw frames—46 trajectories at N=200 (every sequence qualifies) and 40 at N=500 (the six 500-frame *stairs* sequences fall below the 1000-frame raw threshold). For N=2000 no single sequence is long enough, so we concatenate sequences within each scene end-to-end and take the first 2000 sampled frames, yielding 5 scene-level trajectories (*chess, fire, office, pumpkin, redkitchen*; *heads* and *stairs* have insufficient total raw frames after gap-2 subsampling).

Outdoor multi-view sequences. We preserve the source frame order and take the first $N \in \{200, 500, 2000\}$ frames of each sufficiently long trajectory. Oxford Spires [42] provides 14 trajectories at each length, and MatrixCity [20] provides 10 (five aerial and five street-level). The four CrossLoc [51] flight segments contain 1176, 841, 623, and 373 frames, giving four trajectories at $N = 200$, three at $N = 500$, and none at $N = 2000$. CUT3R and TTT3R use the same inputs. We evaluate relative-pose pairs among the final 100 frames and average per-trajectory AUCs without weighting. Pairs with undefined translation direction are excluded from translation AUC in both branches. Both branches share the reconstruction backend over all N frames; POSEPLANER supplies refined pairwise rotations. Table 8 reports the additional tests on CrossLoc drone sequences at $N = 200$ and $N = 500$.

Table 8: **Zero-shot multi-view refinement on CrossLoc drone sequences.** R@5/R@20 and t@5/t@20: rotation/translation-direction AUC at $5^\circ/20^\circ$ (%, ↑). Better results within each pair are **bold**.

Method	$N = 200$				$N = 500$			
	R@5	R@20	t@5	t@20	R@5	R@20	t@5	t@20
CUT3R	**20.5**	55.2	22.6	64.6	11.7	42.7	5.1	45.0
CUT3R + POSEPLANER	20.3	**60.9**	**24.6**	**68.4**	**12.1**	**44.5**	**5.7**	**48.1**
TTT3R	26.0	69.6	**33.3**	**73.5**	20.5	70.9	24.8	73.6
TTT3R + POSEPLANER	**26.2**	**70.0**	29.6	65.9	**25.1**	**73.0**	**28.2**	**75.0**

C.2 External Match-Source Sensitivity

POSEPLANER's geometry tokens $\{\boldsymbol{g}_m\}$ come from patch correspondences produced by an external matcher (Sec. 3.1); this matcher is in principle independent of the upstream pose source. To verify that the refiner's gain is not tied to a particular matcher, we fix the upstream pose source to either SUPERGLUE or LOFTR and sweep four matchers (SUPERGLUE, LOFTR, ROMA, MAST3R) as the source of patch correspondences fed into the diffusion refiner. Tab. 9 reports the resulting two-view rotation AUC on ScanNet1500. POSEPLANER improves over the init pose for the vast majority of (upstream, matcher) combinations, and stronger matchers (ROMA, MAST3R) yield clearly larger gains; the qualitative trend is preserved across all four matchers, confirming that the refiner is robust to the choice of correspondence source.

Table 9: **Sensitivity to the external match source.** Two-view rotation AUC on ScanNet1500 with the upstream pose fixed to SUPERGLUE or LOFTR, while the matcher used to produce patch correspondences for POSEPLANER's conditioning is varied across four options. The *init* row reports each upstream's raw pose accuracy from Tab. 1. Higher is better.

Conditioning match	Upstream: SUPERGLUE			Upstream: LOFTR		
	R@5	R@10	R@20	R@5	R@10	R@20
init (no refinement)	43.9	62.4	76.4	11.5	25.0	44.0
+ POSEPLANER, SUPERGLUE [34] match	45.1	64.1	78.0	17.8	38.8	59.5
+ POSEPLANER, LOFTR [39] match	43.5	62.6	76.8	14.5	31.1	50.5
+ POSEPLANER, ROMA [11] match	**49.8**	**69.6**	**82.5**	**32.9**	**56.3**	**72.8**
+ POSEPLANER, MAST3R [19] match	48.9	68.9	82.1	29.4	52.9	70.6

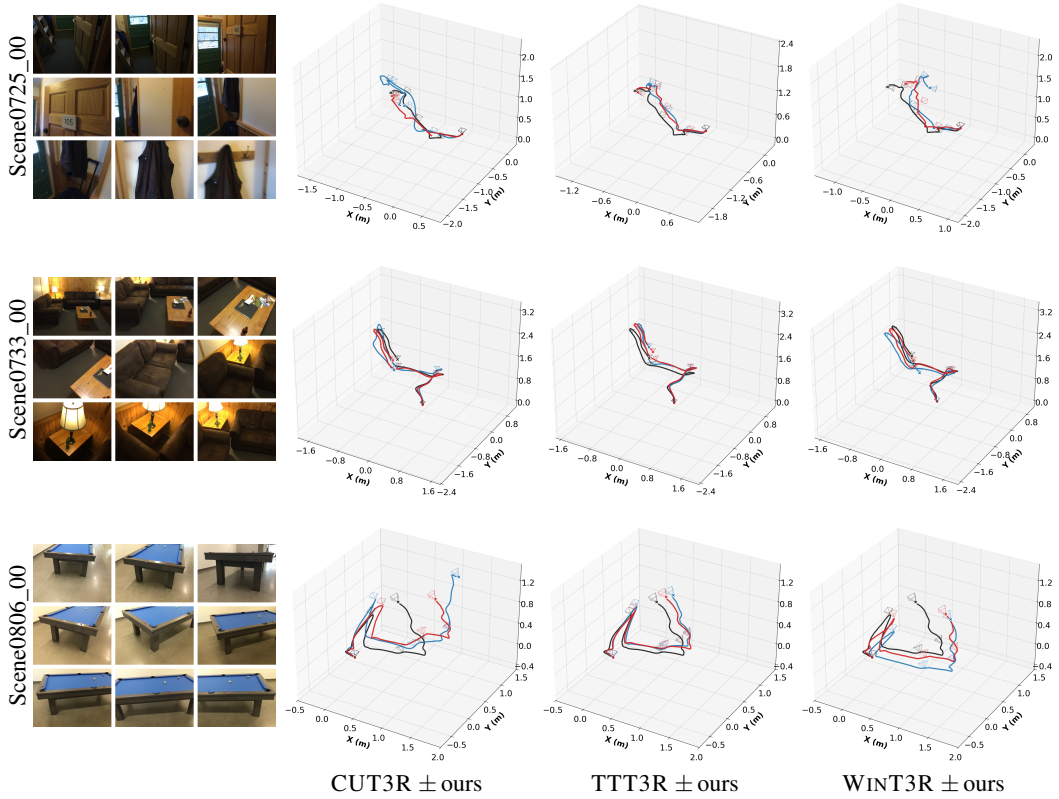


Figure 7: **Additional streaming trajectories on ScanNet.** Each row is one 200-frame sequence refined by POSEPLANER in the same 100-frame sliding-window scheme as Fig. 4 (50-frame overlap). Right three columns plot the estimated camera trajectory under CUT3R, TTT3R, and WINT3R; **black** is ground truth, **blue** is the raw baseline, and **red** is after POSEPLANER refinement.

C.3 Additional Streaming Trajectories

Sliding-window protocol. Each trajectory is 200 frames long. We slide a 100-frame window with 50-frame overlap, so the trajectory contains three refinement windows ($[0, 100)$, $[50, 150)$, $[100, 200)$); within each window, POSEPLANER treats every pair as the two-view problem of Sec. 4.1 and the per-pair refined poses are aggregated by rotation averaging [13, 5] plus a fix-rotation GLOMAP-style positioner [32]. The same protocol is used for the main paper's Fig. 4.

Additional trajectories. Fig. 7 extends the streaming results of Sec. 4.2 with three additional sequences spanning low-texture and cluttered indoor settings; each is refined on top of CUT3R [47], TTT3R [7], and WINT3R [21]. Across all three sequences the refined trajectory tracks the ground truth substantially closer than the raw baseline, confirming that the refiner's drift correction is consistent across upstream estimators rather than tuned to any one of them.

C.4 Training Objective Comparison

We compare flow matching with deterministic Direct-x_0 regression and DDPM pred-x_0, using the same architecture, training data, and 20-epoch schedule. Direct-x_0 predicts the clean pose residual in one forward pass; DDPM uses 100 denoising steps, and flow matching uses one Euler step. All variants use MASt3R correspondences and the same geometric translation solver. Both alternatives improve the LoFTR initialization, but flow matching achieves higher rotation and translation-direction AUC at all reported thresholds (Tab. 10). Each objective is trained once.

Table 10: **Training objective comparison on ScanNet1500.** LoFTR initialization, 1028 valid pairs. R@θ/t@θ: rotation/translation-direction AUC at θ° (%, ↑). Best results are **bold**.

Method	R@5	R@10	R@20	t@5	t@10	t@20
LoFTR init	11.5	25.0	44.0	2.1	5.6	13.0
Direct-x_0	14.8	30.3	49.2	2.3	6.6	14.5
DDPM pred-x_0	16.7	33.2	51.8	2.2	6.5	14.6
Flow matching	**29.4**	**52.9**	**70.6**	**3.5**	**9.0**	**18.4**

C.5 Alternative Plane Source

We replace the learned plane predictions with planes fitted to frozen MoGe [48] point maps using patchwise RANSAC/PCA. The refiner, initial poses, correspondences, and remaining modules are unchanged, with no retraining. Table 11 shows that MoGe-derived planes retain substantial gains over the LoFTR initialization, although our learned planes achieve higher rotation AUC at all three thresholds.

Table 11: **Alternative plane source on ScanNet1500.** Rotation AUC at $5^\circ/10^\circ/20^\circ$ (%, ↑) on the same 1028 valid LoFTR pairs. Best results are **bold**.

Plane source	R@5	R@10	R@20
LoFTR init (no refinement)	11.5	25.0	44.0
MoGe + RANSAC	27.1	50.4	69.2
Learned planes (ours)	**29.4**	**52.9**	**70.6**

D Social Impacts

Positive social impact. POSEPLANER corrects relative camera pose at the geometric level and extends from two-view training to streaming n-view inference without retraining. This makes accurate pose estimation more accessible to downstream applications such as AR/VR, robotics, and indoor scene understanding.

Potential negative impact. As with any technology that improves camera pose estimation, our method could in principle be used in privacy-sensitive contexts. We recommend deploying it within a broader pipeline that applies the appropriate consent and access controls.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The three contributions stated in the abstract and Sec. 1—reviving the correspondence–pose coupling via flow matching, the hybrid SE(3) refinement (flow-matching rotation with score-based fusion plus geometric translation solver), and extending a two-view-trained model to streaming n-view estimators without retraining—are supported by Tab. 1, Tab. 3, and Fig. 4.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Sec. 4.4 discusses both the dependency on planar scene structure and the saturation behaviour on near-ceiling upstream poses.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: All analytical results—the $(\Delta \boldsymbol{n}, \Delta d)$ first-order expansion, the rot6d continuity argument, and the single-step Euler closed form—are derived with explicit assumptions in App. A.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Training recipe and hyperparameters are in App. B.3, architecture details in App. B.1–B.2, and the exact test-pair/sequence construction in App. C.1.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in

some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: All training and evaluation configurations (data mixture, training recipe, pair/sequence construction) are specified in App. B.3, B.1, and C.1; all datasets used are publicly available, and code with pretrained weights will be open-sourced upon acceptance.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Training hyperparameters (optimizer, schedule, resolution, batch size, epochs) are in App. B.3 and test splits are fixed and listed in App. C.1.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: AUC is averaged over thousands of fixed test pairs and replicated across six upstreams and two benchmarks, with deterministic single-step inference; this matches the statistical-reporting convention of the two-view and SLAM pose-estimation literature.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.

- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Total training compute (128 GPU-days on NVIDIA A800 GPUs) and per-stage breakdown are reported in App. B.3.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: All data come from publicly released datasets used under their licenses; no human subjects or personally identifiable data are involved.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

Justification: App. D discusses both the positive impact (lower-cost accurate pose for AR/VR, robotics, indoor SfM/SLAM) and the potential negative impact (privacy risk when combined with downstream indoor 3D reconstruction).

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: The released model only produces a relative camera pose from an RGB pair; it does not generate text, images, or any other synthetic content, and the paper poses no such high-risk-of-misuse scenario.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: All datasets and pretrained models we build upon are properly cited and used under their original licenses.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The trained POSEPLANER checkpoint, training/evaluation code, and exact benchmark pair lists will be open-sourced upon acceptance with documentation and reproducible configurations.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing or research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: The paper does not involve research with human subjects, so no IRB approval is required.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: LLMs are not part of the core method; any usage was limited to writing assistance and does not affect methodology or results.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.