NATIVEREFINE: NUISANCE-AWARE REFINEMENT OF FEED-FORWARD 3D GEOMETRY

Anonymous authors
Paper under double-blind review

ABSTRACT

We introduce NativeRefine, a portable, training-free refinement layer that turns frozen feed-forward 3D predictions into scene-adapted geometry using RGB correspondences. Its key principle is to separate fitting freedom from output complexity. Bounded, centered source-ray variables provide temporary structural flexibility for camera fitting; a second stage fixes the fitted cameras and refits a compact, native-relative depth field. The emitted reconstruction retains native detail and preserves the window-wide geometric mean of depth exactly. Across four frozen backbones and 600 windows from three benchmarks, 2,400 model–window evaluations demonstrate broad applicability without retraining or architecture-specific tuning. With known intrinsics, point and shared-scale translation errors improve together in 10 of 12 cohort–backbone cells against each of native prediction and bundle adjustment with the same dense refit. Two additional recordings tested after method lock retain translation gains of 27.6% and 26.9% over native MapAnything and Pi3X. RGB-only transfer to VGGT and DA3 improves point error in 9 of 12 cells without external calibration. The complete 1,200-evaluation extension runs on a 16 GB consumer GPU, bringing compact, native-preserving optimization to existing reconstruction models.

1 INTRODUCTION

We present **NativeRefine**, a portable, training-free layer that refines dense geometry and cameras using RGB correspondences. Feed-forward models provide the detailed starting representation: DUSt3R and MASt3R connect learned reconstruction with correspondence estimation (Wang et al., 2024; Leroy et al., 2024), while VGGT, MapAnything, and π^3 extend this direction to joint multi-view prediction (Wang et al., 2025; Keetha et al., 2026; Wang et al., 2026). NativeRefine brings complementary, scene-specific image evidence to these predictions through a shared geometric interface, *without retraining or modifying the backbone*.

Our key idea is to *separate fitting freedom from output complexity*. Conventional bundle adjustment gives sparse landmarks freedom to explain image measurements (Triggs et al., 2000; Schönberger & Frahm, 2016). A feed-forward prediction supplies dense structure that a compact residual can preserve, while camera fitting can exploit additional freedom along individual matched rays. NativeRefine assigns these roles to different representations: temporary source-ray adjustments support camera fitting, and a smooth native-relative field expresses the final reconstruction.

Stage 1 jointly updates anchored cameras and a smooth depth field, with one bounded nuisance variable per unique matched source ray. Each ray shares its correction across target views; component-wise centering removes the common log-depth mode. Stage 2 fixes the fitted cameras, discards the nuisance, and refits the persistent depth field to the original correspondences. The output remains dense depth and cameras in the native representation.

This design makes the *refinement* lightweight: low-resolution depth coefficients and small camera increments reuse high-resolution native detail. Window-wide centering preserves the native geometric mean of depth exactly. All network weights remain frozen, with no learned refinement parameters or architecture-specific layers. Fixed intrinsics, native predictions, and RGB matches suffice; ground-truth poses and depths enter only scoring.

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

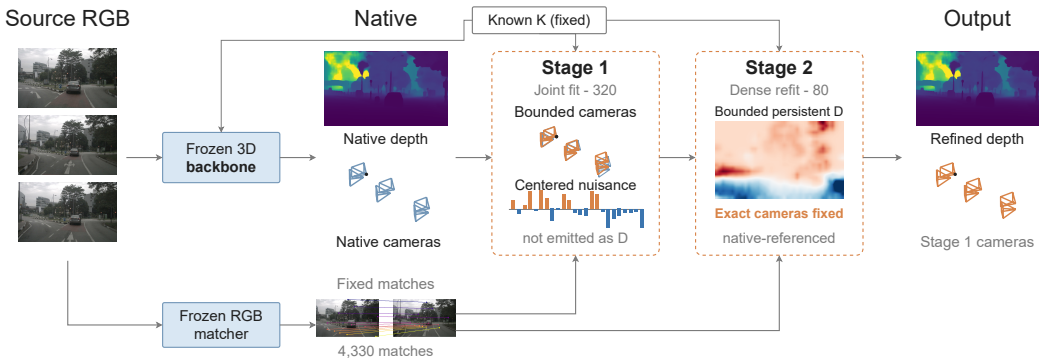

Figure 1: **Fit with temporary freedom; emit constrained geometry.** Frozen RGB matches drive 320 joint updates of cameras, persistent depth, and centered source-ray nuisance. The nuisance is discarded before 80 depth-only updates with exact fitted cameras. The illustrated Known-K MapAnything window uses real RGB, depth, cameras, and signed corrections; three thumbnails summarize six views. Both networks stay frozen. The Predicted-K protocol instead fixes the native model's intrinsics (Table 1).

Our study spans four frozen backbones and the same 600 unique windows of driving, synthetic navigation, and indoor RGB-D data: 2,400 model–window evaluations. Points and cameras share one depth-derived scale, testing dense shape and camera motion together. With known intrinsics, NativeRefine improves both endpoints in 10 of 12 cohort–backbone cells against each of native prediction and native-prior BA with the *same* dense refit. MapAnything TartanAir translation error falls by 43.6% on DEV and 43.8% on CAL against this matched control. Two further recordings tested after method lock retain translation gains of 27.6% and 26.9% over native MapAnything and Pi3X. A separate RGB-only protocol extends the fixed refiner to VGGT and DA3 with predicted intrinsics, improving point error in 9 of 12 cells over native. All 1,200 extension evaluations complete on one 16 GB consumer GPU through serial backbone, matching, and refinement stages.

Our contributions are threefold. **A fitting–emission separation** couples temporary source-ray freedom with persistent dense reconstruction. **A compact, native-preserving parameterization** combines centered nuisance variables, bounded camera updates, and an exact depth statistic. **A four-backbone study with paired mechanism controls** demonstrates calibrated point–camera gains, transfer to additional recordings, and RGB-only compatibility with one training-free design.

2 Related Work

Geometric optimization with learned representations. Classical bundle adjustment jointly estimates cameras and structure from image measurements (Triggs et al., 2000; Schönberger & Frahm, 2016). CodeSLAM and DeepFactors optimize compact learned depth representations with camera motion (Bloesch et al., 2018; Czarnowski et al., 2020); BA-Net embeds dense bundle adjustment in a learned system (Tang & Tan, 2019). DeepV2D, DROID-SLAM, and Deep Patch Visual Odometry combine learned geometric evidence with iterative depth or structure and pose estimation (Teed & Deng, 2020; 2021; Teed et al., 2023). MegaSaM adapts this family to difficult casual videos (Li et al., 2025). NativeRefine brings this optimization perspective to an already dense frozen prediction: sparse source-ray variables provide fitting flexibility, while a compact correction preserves the native dense scaffold for output. Its BA–refit control compares this coupled design with cameras obtained from conventional free-landmark optimization.

Depth consistency and foundation-model refinement. VGGT and Depth Anything 3 infer multi-view geometry and cameras from RGB (Wang et al., 2025; Lin et al., 2025), enabling refinement with predicted rather than externally supplied intrinsics. Consistent Video Depth Estimation adapts depth predictions to multi-view constraints, while Robust Consistent Video Depth Estimation uses low-frequency deformation to improve consistency (Luo et al., 2020; Kopf et al., 2021). DUSt3R, MASt3R-SfM, and MonST3R use geometric optimization around learned predictions (Wang et al.,

2024; Duisterhof et al., 2025; Zhang et al., 2025). Most directly, EPO refines foundation-model cameras, focal lengths, and depth through edge alignment (D'Urso et al., 2026). It first fits cameras and intrinsics, then includes depth in joint optimization, using pose-MLP and pixel-wise affine depth updates. NativeRefine couples centered transient and persistent depth spaces under fixed point matches and calibration. Temporary source-ray freedom supports joint fitting; a subsequent *fixed-camera* refit transfers the fitted dense coefficients into an objective defined directly on the output representation. This representation and transfer rule separates camera-fitting flexibility from compact dense prediction.

Correspondences, parameterization, and evaluation. MASt3R grounds matching in learned 3D geometry; RoMa and RoMa v2 provide robust dense RGB matching (Leroy et al., 2024; Edstedt et al., 2024; 2025). Frozen RoMa v2 correspondences give both NativeRefine stages a common set of image measurements. Projection conditioning is a separate issue from correspondence quality: singularity-free bundle adjustment explicitly addresses the former (Garcia, 2020). Our evaluation measures depth shape and camera errors separately to resolve their response to refinement. Driving imagery, simulated navigation, and indoor RGB-D benchmarks provide complementary observation regimes (Caesar et al., 2020; Wang et al., 2020; Sturm et al., 2012), with depth and pose supervision reserved for scoring.

3 Correspondence-Guided Bounded Refinement

NativeRefine separates the freedom useful for camera fitting from the representation required for dense output. Stage 1 combines a compact dense correction with temporary source-ray depth adjustments; Stage 2 holds the fitted cameras fixed and refits the dense correction alone. Both stages use the same image correspondences and native geometric reference.

Inputs and fixed image evidence. A window contains V RGB images of size $H \times W$, indexed by $i = 0, \dots, V-1$, with fixed intrinsics $K_i \in \mathbb{R}^{3\times 3}$, positive native camera-z depths D_i^0, and native camera-to-world transforms $T_i^0 \in \mathrm{SE}(3)$. Intrinsics are supplied calibration in the Known-K protocol or native estimates in the Predicted-K protocol; neither stage updates them. For any pose T_i, let R_i and C_i denote its rotation and camera-center blocks. The backbone is frozen. RoMa v2 supplies bidirectional matches between adjacent views (Edstedt et al., 2025). We sample source pixels at stride eight beginning at $(4, 4)$, retain finite in-bounds target coordinates with overlap at least 0.5 and cycle error at most 1.5 pixels, and keep at most 1024 evenly spaced matches per direction. This produces $\mathcal{M} = \{(i, j, p, q, w)\}$, where $p = (p_x, p_y)$ and $q \in \mathbb{R}^2$ are source and target pixels, i, j their views, and w the overlap score. Match selection uses neither predicted depth nor ground truth. The matches, weights, target coordinates, and intrinsics are fixed in both stages.

3.1 Persistent bounded depth and camera state

We refine native depth multiplicatively, retaining its dense spatial detail while adjusting a low-resolution log-depth field. Write $a = (a_i)_{i=0}^{V-1}$ for all raw depth coefficients. Each a_i lies on a $\lfloor H/14 \rfloor \times \lfloor W/14 \rfloor$ grid. Let $\langle\cdot\rangle_c$ average over all coarse cells and views in the window, let $\langle\cdot\rangle_f$ average over all full-resolution pixels and views, and let B be bilinear interpolation with `align_corners=False`. We decode

$$c(a) = 0.175\big(\tanh a - \langle \tanh a\rangle_c\big),$$
$$\delta(a) = Bc(a) - \langle Bc(a)\rangle_f, \qquad D_i(a,p) = D_i^0(p)\exp\delta_i(a,p). \tag{1}$$

This is the persistent depth field optimized and emitted in both stages. For $N_{\mathrm{pix}} = VHW$, its exact window-level invariant is

$$\frac{1}{N_{\mathrm{pix}}}\sum_{i,p}\log\frac{D_i(a,p)}{D_i^0(p)} = 0. \tag{2}$$

Indeed, the second centering in Eq. (1) makes the mean log ratio exactly zero, preserving the native geometric-mean depth. Bilinear interpolation preserves constants and forms convex combinations; hence $\delta = 0.175(B\tanh a - \langle B\tanh a\rangle_f)$ and $|\delta_i(p)| \le 0.35$. Depth therefore stays positive and its multiplicative change lies in $[e^{-0.35}, e^{0.35}]$. The native field thus supplies the reference geometric-mean scale, while the correction redistributes relative depth within this bounded range.

Camera updates use the same bounded right-composed decoder. With $m = \mathrm{median}_{i,p}\, D_i^0(p)$ fixed for the window (float32 lower median for an even number of values) and raw $b_i \in \mathbb{R}^6$,

$$\omega_i = 0.05 \tanh b_{i,1:3}, \qquad v_i = 0.02m \tanh b_{i,4:6}, \qquad T_i = T_i^0 \exp\begin{pmatrix} [\omega_i]_\times & v_i \\ 0 & 0 \end{pmatrix}. \tag{3}$$

Here $[\omega]_\times x = \omega \times x$, and exp of a matrix denotes the matrix exponential. We form the undirected view graph induced by $\mathcal{M}$ and fix the smallest-index view in every connected component, including isolated views, to its native pose. Each component thereby retains a native coordinate reference by setting its $b_i = 0$. The vector v_i is the local translation coordinate in the right-composed Lie-algebra update; $b = (b_i)_{i=0}^{V-1}$ denotes the full camera state.

3.2 Stage 1: nuisance-aware joint refinement

During camera fitting, temporary source-ray adjustments provide local depth flexibility alongside the compact persistent field. Repeated directed matches can share the same integer source ray. Let $\mathcal{U}$ be the lexicographically ordered unique source keys $s = (i, p_x, p_y)$, and let $g(s)$ denote the view-graph component containing the source view. One raw scalar η_s is shared by every match with key s, so its target matches constrain one shared source depth. For each nonempty set $\mathcal{U}_g = \{s \in \mathcal{U} : g(s) = g\}$ and $s \in \mathcal{U}_g$, we decode

$$r_s = \tanh \eta_s, \qquad \mu_g = \frac{1}{|\mathcal{U}_g|} \sum_{s \in \mathcal{U}_g} r_s, \qquad \ell_s = 0.7\, \frac{r_s - \mu_g}{1 + |\mu_g|}. \tag{4}$$

The denominator is common within a component, so $\sum_{s \in \mathcal{U}_g} \ell_s = 0$; because $r_s, \mu_g \in [-1, 1]$, it also ensures $|\ell_s| \le 0.7$. Centering allocates this transient freedom to relative source-ray adjustments: the common log-depth mode within each component is excluded exactly. Every unique source ray contributes equally to this constraint, regardless of its number of target matches. Collect the variables as $\eta = (\eta_s)_{s \in \mathcal{U}}$ and $\ell = (\ell_s)_{s \in \mathcal{U}}$.

For $(i, j, p, q, w) \in \mathcal{M}$, define the target-camera point for any depth field D, cameras T, and nuisance ℓ as

$$X_{ij}(p; D, T, \ell) = \left[T_j^{-1} T_i\right]_{1:3,:} \begin{bmatrix} D_i(p) e^{\ell_{(i,p_x,p_y)}} K_i^{-1} \bar{p} \\ 1 \end{bmatrix}, \qquad \widehat{q} = \pi_{K_j}(X_{ij}), \tag{5}$$

where $\bar{p} = (p_x, p_y, 1)^\top$ and $[\cdot]_{1:3,:}$ selects the first three matrix rows. Projection is $\pi_K(X) = (KX)_{1:2}/d_\epsilon(X_z)$, where $d_\epsilon(z) = \max(z, \epsilon)$ for $z \ge 0$ and $\min(z, -\epsilon)$ otherwise, with $\epsilon = 10^{-6}$. We suppress $(p; D, T, \ell)$ in X_{ij} below. Let $s_I = \max(H, W)$ and define Huber loss by $\rho_\beta(r) = r^2/2$ for $|r| \le \beta$ and $\beta(|r| - \beta/2)$ otherwise, with $\beta = 0.005$. The shared correspondence objective is

$$\mathcal{L}_{\mathrm{corr}}(D, T, \ell) = \frac{\sum_{\mathcal{M}} w \sum_{k=1}^{2} \rho_{0.005}\big((\widehat{q}_k - q_k)/s_I\big)}{\sum_{\mathcal{M}} w} + \frac{1}{|\mathcal{M}|} \sum_{\mathcal{M}} \left[\max\big(0, 10^{-4} - X_{ij,z}/m\big)\right]^2. \tag{6}$$

Thus overlap weights normalize only the coordinate term; the cheirality term is an unweighted mean over directed matches.

The pose prior acts on decoded camera centers C_i, with native centers C_i^0. Set $\sigma_C = 0.02\, \mathrm{median}_{64}(D^0)$, where D^0 collects all native depths and the median uses float64 with even-length averaging. Then

$$\mathcal{L}_C = \frac{1}{2|\mathcal{M}|s_I^2} \sum_{i=1}^{V-1} \left\| \frac{C_i - C_i^0}{\sigma_C} \right\|_2^2. \tag{7}$$

Component anchors contribute zero; rotations are controlled by the bounded decoder and correspondence term. Define $\mathcal{G}(c) = \frac{1}{2}(\langle(\Delta_x c)^2\rangle_x + \langle(\Delta_y c)^2\rangle_y)$, where Δ_x, Δ_y are adjacent-cell forward differences and each mean covers its valid coarse-grid edges and all views. Let $\langle\cdot\rangle_{\mathcal{U}}$ average over unique source rays. Stage 1 minimizes

$$\mathcal{L}_1 = \mathcal{L}_{\mathrm{corr}}(D(a), T(b), \ell) + 0.01\langle c(a)^2\rangle_c + 0.01\mathcal{G}(c(a)) + \mathcal{L}_C + 10^{-6}\langle \ell^2\rangle_{\mathcal{U}}. \tag{8}$$

All raw variables start at zero. Adam jointly updates a, the non-anchor camera variables b, and η for 320 fixed steps at learning rate 0.03.

Table 1: Four-backbone input protocols. Each refiner uses one fixed K per view; GT poses never enter inference. Known-K and Predicted-K results are reported separately.

Backbone	Native input	Refinement K	GT pose input
MapAnything	RGB + Known K	Known, fixed	None
Pi3X	RGB	Known, fixed	None
VGGT-1B	RGB	Predicted, fixed	None
DA3-GIANT-1.1	RGB	Predicted, fixed	None

3.3 Stage 2: fixed-camera dense refit

Let $(a^{(1)}, T^{(1)}, \ell^{(1)})$ denote the completed Stage 1 state. We copy $a^{(1)}$, freeze $T^{(1)}$ exactly, and run 80 updates with a newly initialized Adam optimizer at learning rate 0.03 on depth coefficients alone. The ordinary emitted reprojection sets the source-ray nuisance to zero:

$$\mathcal{L}_2(a; T^{(1)}) = \mathcal{L}_{\text{corr}}(D(a), T^{(1)}, 0) + 0.01\langle c(a)^2\rangle_c + 0.01\mathcal{G}(c(a)). \tag{9}$$

Writing $a^{(2)}$ for the coefficients after these 80 updates, the final output is $(D(a^{(2)}), T^{(1)})$. The warm start carries the coarse correction learned during joint fitting into the dense refit. Removing the source-ray variables makes this second objective operate directly on the emitted depth, with camera fitting already complete. Stage 1 nuisance values may be retained for diagnostics after transfer.

Both stages use only inference inputs; predictions are saved before scoring. Appendix B specifies the recorded native-fallback conditions.

4 Experiments

4.1 Evaluation design

We evaluate four frozen backbones on the same 600 input windows: 2400 model–window evaluations across driving, synthetic navigation, and real indoor RGB-D. The **Known-K** study uses MapAnything and Pi3X (Keetha et al., 2026; Wang et al., 2026); both refiners receive calibration, while only MapAnything receives it during native inference. The **Predicted-K** extension uses RGB-only VGGT and DA3 (Wang et al., 2025; Lin et al., 2025), retaining each model's predicted intrinsics throughout refinement. No protocol supplies target poses. Each native prediction and fixed RoMa v2 matches are shared across its controls. The same losses, bounds, weights, and 320+80 schedule apply throughout, without network training or backbone-specific tuning.

Coverage and aggregation. DEV comprises 18 nuScenes windows from six logs and 256 TartanAir windows from one environment (Caesar et al., 2020; Wang et al., 2020). Previously evaluated cohorts contain 15 nuScenes windows from four logs, 64 TartanAir windows from two environments, and 155 TUM windows from three recordings (Sturm et al., 2012). Two further TUM recordings, `nostructure_texture_far` and `teddy`, contribute 15 and 77 windows after method lock. *Fresh* denotes their first use for refiner evaluation. Means weight physical groups equally, giving each fresh recording half the weight. Each protocol has twelve cohort–backbone cells. NuScenes uses six views per window; TUM and TartanAir use three. Appendix C gives sampling and support.

Joint geometry and matched controls. Primary endpoints are shared-scale point-relative error (P) and camera-center error (ST): one median target-to-predicted depth ratio scales both predicted points and translations. Prediction and target points use their respective intrinsics; Known-K makes them identical. Ground truth enters only scoring. Rotation and independently scale-aligned (SA) translation provide complementary camera diagnostics. **BA+refit** fits cameras and independent two-observation landmarks from the same RGB matches, with fixed intrinsics, component anchors, and native-center priors, then applies the same 80-step dense refit to native depth. Every prescribed window is scored, with zero recorded native fallbacks. Appendices A and D define scoring and controls.

Table 2: NativeRefine improves both shared-scale errors in 10/12 cohort–model cells versus native and in 10/12 versus BA+refit. Point error is relative; translation uses the same fitted scale and is reported in cm. Equal physical-group means give far and teddy equal weight in fresh TUM. Bold marks the best displayed values, including ties.

Cohort	Model	Shared-scale point relative ↓			Shared-scale T (cm) ↓		
		Native	BA+refit	Ours	Native	BA+refit	Ours
nuScenes DEV	MapAnything	0.0903	**0.0873**	0.0876	53.10	**41.33**	44.76
nuScenes DEV	Pi3X	0.0623	**0.0606**	0.0608	35.81	**28.31**	30.63
TartanAir DEV	MapAnything	0.2761	0.2738	**0.2721**	5.59	3.41	**1.92**
TartanAir DEV	Pi3X	0.1750	0.1750	**0.1737**	2.55	1.85	**1.40**
nuScenes CAL	MapAnything	0.1668	0.1660	**0.1657**	28.73	30.83	**26.96**
nuScenes CAL	Pi3X	**0.0898**	0.0900	**0.0898**	17.19	20.13	**15.85**
TartanAir CAL	MapAnything	0.1791	0.1773	**0.1742**	8.54	5.81	**3.27**
TartanAir CAL	Pi3X	0.0707	0.0706	**0.0700**	3.12	**2.78**	**2.78**
TUM observed	MapAnything	0.0445	0.0478	**0.0433**	1.91	1.58	**1.34**
TUM observed	Pi3X	0.0289	0.0315	**0.0288**	**0.77**	0.88	0.80
TUM fresh	MapAnything	0.1328	0.1402	**0.1314**	4.77	3.59	**3.45**
TUM fresh	Pi3X	0.0680	0.0715	**0.0667**	3.05	2.81	**2.23**

4.2 Cross-domain reconstruction with known intrinsics

NativeRefine lowers P and ST individually in 11 of 12 cells relative to native, and both together in ten (Table 2). Both errors also improve over BA+refit in ten cells, including *all eight non-DEV cells*. The gains transfer across the two backbones and the tested driving, synthetic, and real indoor domains with one fixed configuration.

Better depth–motion coupling. On MapAnything TartanAir, shared-scale translation falls from 3.41 to 1.92 cm on DEV and 5.81 to 3.27 cm on CAL relative to BA+refit: reductions of 43.6% and 43.8%. Reductions relative to native reach 65.7% and 61.7%. Point error also improves over BA+refit by 0.61% and 1.75%. Pi3X improves both endpoints on both splits, including 24.6% lower translation than BA+refit on DEV. The matched dense refit makes these comparisons tests of the final camera–depth pair.

Fixed-method transfer to additional recordings. Fresh TUM shared-scale translation improves by 27.6% for MapAnything and 26.9% for Pi3X over native; point error improves by 1.06% and 1.92%. Against BA+refit, point error is 6.28% and 6.63% lower, and translation is 3.78% and 20.64% lower. Appendix Table 6 reports both recordings separately. These results demonstrate transfer without refiner retraining or retuning.

4.3 Camera accuracy and distributed gains

NativeRefine reduces independently aligned translation relative to native in 11 of 12 Known-K cells and surpasses BA on all four TartanAir cells (Table 3). On nuScenes, it also has lower rotation error than BA for both models and splits. MapAnything shared-scale translation improves in 318/320 TartanAir windows; even the 95th-percentile change is favorable on both splits. Fresh TUM median translation changes are negative for both backbones (Appendix Table 14).

External refinement comparison. Against official EPO edge-based refinement (D'Urso et al., 2026), NativeRefine has lower rotation and SA translation in all ten historical Known-K cells under their common original rotation scorer. EPO optimizes focal length and depth under its official convergence settings; BA+refit supplies the matched fixed-K dense comparison. Appendix D and Table 7 report the protocols and complete EPO results.

4.4 Backbone portability with predicted intrinsics

The RGB-only extension changes neither the refiner nor its settings. NativeRefine lowers point error in 9/12 cells versus native and 10/12 versus BA+refit (Table 4). VGGT nuScenes DEV improves

Table 3: Camera metrics separate rotation accuracy from scale-aligned translation accuracy across the twelve cohort–model cells. Rotation projects predicted and GT relative rotations to SO(3); SA T denotes independently scale-aligned translation. Lower is better; best displayed values and ties are bold.

Cohort	Model	SO(3) rotation (°) ↓			SA T (cm) ↓		
		Native	BA+refit	Ours	Native	BA+refit	Ours
nuScenes DEV	MapAnything	1.405	1.147	**0.917**	40.13	**26.31**	28.87
nuScenes DEV	Pi3X	1.111	1.080	**0.784**	30.13	**20.15**	21.82
TartanAir DEV	MapAnything	0.456	**0.022**	0.137	3.77	0.83	**0.65**
TartanAir DEV	Pi3X	0.158	**0.021**	0.064	1.49	0.39	**0.32**
nuScenes CAL	MapAnything	1.196	1.241	**0.791**	22.16	24.09	**18.76**
nuScenes CAL	Pi3X	0.668	1.349	**0.619**	14.32	16.75	**12.59**
TartanAir CAL	MapAnything	0.496	**0.118**	0.179	5.07	1.93	**1.44**
TartanAir CAL	Pi3X	0.169	**0.120**	0.150	2.17	1.37	**1.16**
TUM observed	MapAnything	0.736	**0.537**	0.562	1.28	0.87	**0.75**
TUM observed	Pi3X	**0.387**	0.538	0.414	**0.64**	0.75	0.66
TUM fresh	MapAnything	1.345	**0.721**	1.042	3.24	**1.55**	1.87
TUM fresh	Pi3X	**0.616**	0.724	0.666	1.82	1.44	**1.33**

Table 4: Predicted-K transfer without backbone-specific tuning: NativeRefine reduces shared-scale point error in 9/12 cohort–model cells versus native and 10/12 versus BA+refit. Each backbone uses all 600 frozen windows and the same 320+80 updates. Equal physical-group means; far and teddy have equal weight in fresh TUM. Bold: best displayed values and ties.

Cohort	Model	Shared-scale point relative ↓			Shared-scale T (cm) ↓		
		Native	BA+refit	Ours	Native	BA+refit	Ours
nuScenes DEV	VGGT	0.1075	0.1052	**0.1047**	49.53	49.27	**46.88**
nuScenes DEV	DA3	0.1071	0.1045	**0.1042**	**39.25**	45.13	40.48
TartanAir DEV	VGGT	**0.3116**	0.3117	0.3119	**8.23**	8.33	8.79
TartanAir DEV	DA3	0.2190	0.2189	**0.2186**	0.59	**0.46**	0.95
nuScenes CAL	VGGT	0.1475	0.1475	**0.1472**	19.38	38.23	**19.04**
nuScenes CAL	DA3	0.1443	**0.1432**	**0.1432**	**24.15**	25.05	24.55
TartanAir CAL	VGGT	0.2191	0.2194	**0.2190**	9.62	**9.54**	10.65
TartanAir CAL	DA3	0.1066	0.1070	**0.1058**	**1.02**	1.46	2.15
TUM observed	VGGT	**0.0869**	0.0878	0.0871	**1.16**	1.26	1.26
TUM observed	DA3	**0.0409**	0.0428	0.0410	**0.70**	0.86	0.86
TUM fresh	VGGT	0.1665	0.1699	**0.1659**	6.64	6.73	**6.48**
TUM fresh	DA3	0.1301	0.1315	**0.1292**	**3.54**	3.60	3.60

points, shared translation, and rotation by 2.53%, 5.36%, and 1.19% over native. On fresh TUM, full refinement improves all three metrics over the strict no-nuisance+refit control for both new backbones, supporting a useful contribution from temporary fitting freedom (Appendix G).

Dense geometry benefits most consistently from this RGB-only transfer, while camera effects vary across backbones and cohorts. The two input protocols retain separate point and camera results, with complete new-backbone controls in Appendix G.

4.5 Why the two-stage representation helps

The Known-K DEV factorial crosses uncentered versus component-centered nuisance and Stage 1 output versus the additional dense refit. Both nuisance variants retain bounded source-ray freedom and its quadratic penalty. With transfer enabled, centering reduces TartanAir shared translation from 2.40 to 1.92 cm for MapAnything and from 2.29 to 1.40 cm for Pi3X, while improving point scores. This isolates how temporary freedom is organized; the persistent dense field is centered in both arms.

With centered fitting, fixed-camera refitting lowers point error on all four DEV cohorts. TartanAir shared translation also improves with cameras bitwise unchanged, connecting dense correction to the

Table 5: The DEV 2×2 ablation separates nuisance centering from warm depth transfer. All arms use 320 joint updates; transfer adds 80 depth updates. P is shared-scale point-relative error; ST is shared-scale translation in cm. Results are equal-group means under the same four-arm comparison in every cohort.

Cohort	Model	Soft joint		Soft transfer		Centered joint		Centered transfer	
		P ↓	ST ↓	P ↓	ST ↓	P ↓	ST ↓	P ↓	ST ↓
nuScenes DEV	MapAnything	0.0903	44.14	0.0884	45.28	0.0902	**43.48**	**0.0876**	44.76
nuScenes DEV	Pi3X	0.0624	29.39	0.0611	30.26	0.0624	**29.38**	**0.0608**	30.63
TartanAir DEV	MapAnything	0.2750	2.49	0.2730	2.40	0.2746	1.98	**0.2721**	**1.92**
TartanAir DEV	Pi3X	0.1751	2.41	0.1751	2.29	0.1742	1.46	**0.1737**	**1.40**

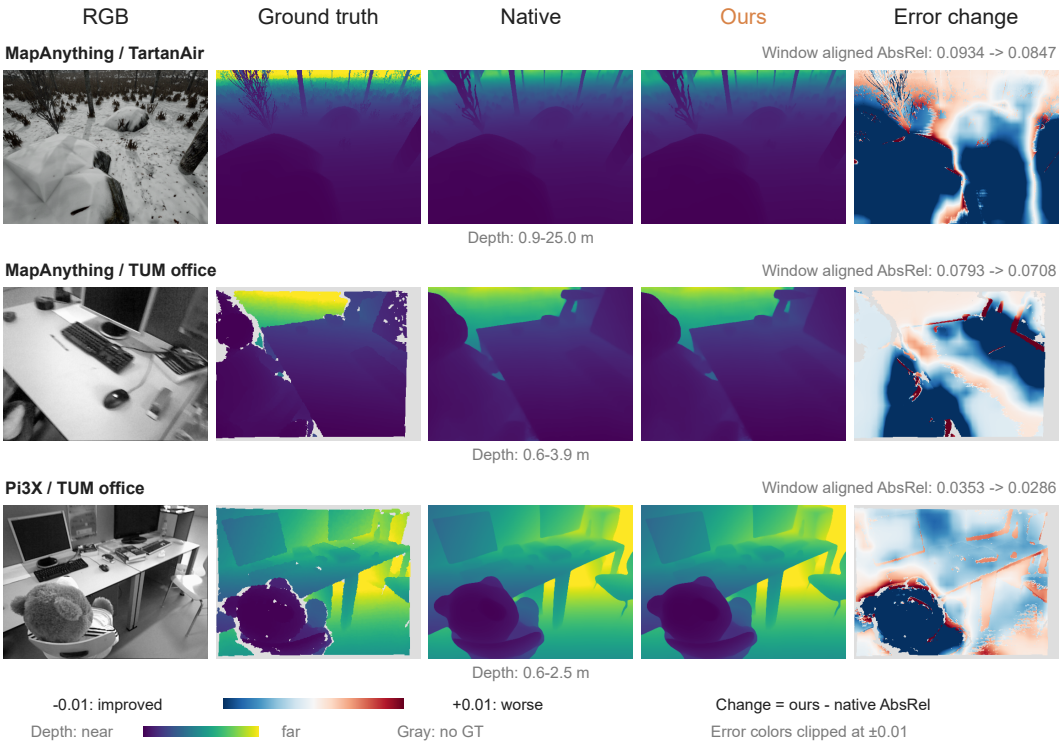

Figure 2: **Dense corrections retain native structure.** TartanAir and previously observed TUM RGB-D examples. Each prediction uses its own full-window depth scale; depth panels share the shown view's target 2nd–98th percentile range. Error change is refined minus native AbsRel, clipped at ± 0.01: blue improves, red worsens, and gray denotes missing targets. Clipping affects display only. Labels report full-window aligned AbsRel.

common reconstruction scale. Together, centered fitting and fixed-camera refitting connect flexible correspondence fitting with a compact, native-preserving output.

Direct mechanism controls. A separate paired replay covers both complete Known-K DEV rosters: 548 backbone–window evaluations of strict no-nuisance fitting and direct nuisance emission (Appendix I). Against strict no-nuisance fitting, temporary ray freedom reduces TartanAir rotation error by 45.4%/28.0% and SA translation by 60.9%/49.9% for MapAnything/Pi3X. On nuScenes, constrained emission with refitting lowers point error by 10.2%/19.7% and shared translation by 17.5%/36.2% relative to direct emission. Refitting improves point error in all four replay cells. The complete controls distinguish camera-fitting gains from depth-scale effects and characterize the backbone-specific output-policy tradeoffs.

Figure 2 shows broad surface corrections alongside detailed native boundaries. The smooth residual acts on high-resolution native depth; the displayed output has already discarded its temporary source-ray variables.

4.6 Compact optimization and practical deployment

NativeRefine adds no trained network. At 336×448, a three-view window uses only 2304 persistent depth coefficients, versus 451584 per-pixel depth values: a $196\times$ reduction in depth-variable count. Camera state adds six variables per non-anchor view, and source-ray nuisance is temporary. The compact optimization reuses native detail instead of differentiating through either frozen network.

On fresh TUM, 320+80 optimization averages 2.69 s for MapAnything and 4.50 s for Pi3X on an RTX A6000; dense refitting contributes only 0.31/0.53 s, approximately 12%. Native inference and matching add 0.26+0.62 s and 0.29+0.65 s, respectively (Appendix Table 15). The complete 1200-evaluation VGGT/DA3 extension runs on one 16 GB RTX 5070 Ti by releasing model weights between stages. Its full-refiner window means are 7.18 s for VGGT and 7.82 s for DA3 across all 600 windows each. Appendix H separates native, matching, refinement, and measured tensor-allocation peaks. The additional 548 Known-K mechanism replays also complete on this consumer GPU, extending local execution evidence to both original backbones.

5 Discussion and Conclusion

NativeRefine turns frozen learned geometry into a reusable prior for scene-specific refinement. Temporary source-ray freedom supports camera fitting; fixed-camera refitting produces a compact correction that preserves the native depth statistic exactly. Nuisance fitting improves TartanAir rotation and independently aligned translation in paired DEV controls; dense refitting lowers point error from Stage 1 in all four evaluated cells.

The calibrated study delivers joint point–camera gains in 10 of 12 cells against each baseline, with roughly 27% translation gains on two further recordings. Four-backbone compatibility and the complete RGB-only extension on a 16 GB consumer GPU establish a practical, training-free design. The central contribution is a portable separation of fitting flexibility and native-preserving output: *refine the geometry, retain the learned representation*.

Reproducibility Statement

Section 3 gives the objectives, decoders, and fixed update schedule. Appendices A–D specify scoring, data preparation, and baselines; Appendices G and I describe RGB-only transfer and paired mechanism controls. The supplement provides the Known-K cached-input refiner, tests, and full-precision evidence with table reproduction for both protocols and the new controls. Real-data reruns use the separately distributed benchmark data and pretrained models.

AI use statement

Generative AI tools assisted with code development and final language polishing.

References

Michael Bloesch, Jan Czarnowski, Ronald Clark, Stefan Leutenegger, and Andrew J. Davison. CodeSLAM—learning a compact, optimisable representation for dense visual SLAM. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. URL https://openaccess.thecvf.com/content_cvpr_2018/html/Bloesch_CodeSLAM_--_Learning_CVPR_2018_paper.html.

Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuScenes: A multimodal dataset for autonomous driving. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. URL https://openaccess.thecvf.com/content_CVPR_2020/html/Caesar_nuScenes_A_Multimodal_Dataset_for_Autonomous_Driving_CVPR_2020_paper.html.

Jan Czarnowski, Tristan Laidlow, Ronald Clark, and Andrew J. Davison. DeepFactors: Real-time probabilistic dense monocular SLAM. *IEEE Robotics and Automation Letters*, 5:721–728, 2020. doi: 10.1109/LRA.2020.2965415. URL https://arxiv.org/abs/2001.05049.

Bardienus Pieter Duisterhof, Lojze Zust, Philippe Weinzaepfel, Vincent Leroy, Yohann Cabon, and Jérôme Revaud. MASt3R-SfM: A fully-integrated solution for unconstrained structure-from-motion. In *International Conference on 3D Vision (3DV)*, 2025. URL https://arxiv.org/abs/2409.19152.

Mattia D'Urso, Christian Sormann, Mattia Rossi, and Friedrich Fraundorfer. Boosting 3D foundation models with edge-based pose optimization. In *European Conference on Computer Vision (ECCV)*, 2026. URL https://arxiv.org/abs/2607.00579.

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. URL https://openaccess.thecvf.com/content/CVPR2024/html/Edstedt_RoMa_Robust_Dense_Feature_Matching_CVPR_2024_paper.html.

Johan Edstedt, David Nordström, Yushan Zhang, Georg Bökman, Jonathan Astermark, Viktor Larsson, Anders Heyden, Fredrik Kahl, Mårten Wadenbäck, and Michael Felsberg. RoMa v2: Harder better faster denser feature matching. *arXiv preprint arXiv:2511.15706*, 2025. URL https://arxiv.org/abs/2511.15706.

E. Garcia. Singularity-free bundle adjustment. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, V-2-2020:43–48, 2020. doi: 10.5194/isprs-annals-V-2-2020-43-2020. URL https://isprs-annals.copernicus.org/articles/V-2-2020/43/2020/.

Nikhil Keetha, Norman Müller, Johannes Schönberger, Lorenzo Porzi, Yuchen Zhang, Tobias Fischer, Arno Knapitsch, Duncan Zauss, Ethan Weber, Nelson Antunes, Jonathon Luiten, Manuel Lopez-Antequera, Samuel Rota Bulò, Christian Richardt, Deva Ramanan, Sebastian Scherer, and Peter Kontschieder. MapAnything: Universal feed-forward metric 3D reconstruction. In *International Conference on 3D Vision (3DV)*, 2026. URL https://arxiv.org/abs/2509.13414.

Johannes Kopf, Xuejian Rong, and Jia-Bin Huang. Robust consistent video depth estimation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. URL https://openaccess.thecvf.com/content/CVPR2021/html/Kopf_Robust_Consistent_Video_Depth_Estimation_CVPR_2021_paper.html.

Vincent Leroy, Yohann Cabon, and Jérôme Revaud. Grounding image matching in 3D with MASt3R. In *European Conference on Computer Vision (ECCV)*, 2024. URL https://arxiv.org/abs/2406.09756.

Zhengqi Li, Richard Tucker, Forrester Cole, Qianqian Wang, Linyi Jin, Vickie Ye, Angjoo Kanazawa, Aleksander Holynski, and Noah Snavely. MegaSaM: Accurate, fast, and robust structure and motion from casual dynamic videos. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. doi: 10.1109/CVPR52734.2025.00981. URL https://arxiv.org/abs/2412.04463.

Haotong Lin, Sili Chen, Jun Hao Liew, Donny Y. Chen, Zhenyu Li, Guang Shi, Jiashi Feng, and Bingyi Kang. Depth anything 3: Recovering the visual space from any views. *arXiv preprint arXiv:2511.10647*, 2025. URL https://arxiv.org/abs/2511.10647.

Xuan Luo, Jia-Bin Huang, Richard Szeliski, Kevin Matzen, and Johannes Kopf. Consistent video depth estimation. *ACM Transactions on Graphics*, 2020. URL https://roxanneluo.github.io/Consistent-Video-Depth-Estimation/.

Johannes L. Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. URL https://openaccess.thecvf.com/content_cvpr_2016/html/Schonberger_Structure-From-Motion_Revisited_CVPR_2016_paper.html.

Jürgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers. A benchmark for the evaluation of RGB-D SLAM systems. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 573–580, 2012. doi: 10.1109/IROS.2012.6385773. URL https://cvg.cit.tum.de/data/datasets/rgbd-dataset.

Chengzhou Tang and Ping Tan. BA-Net: Dense bundle adjustment networks. In *International Conference on Learning Representations (ICLR)*, 2019. URL https://openreview.net/forum?id=B1gabhRcYX.

Zachary Teed and Jia Deng. DeepV2D: Video to depth with differentiable structure from motion. In *International Conference on Learning Representations (ICLR)*, 2020. URL https://arxiv.org/abs/1812.04605.

Zachary Teed and Jia Deng. DROID-SLAM: Deep visual SLAM for monocular, stereo, and RGB-D cameras. In *Advances in Neural Information Processing Systems*, 2021. URL https://proceedings.neurips.cc/paper/2021/file/89fcd07f20b6785b92134bd6c1d0fa42-Paper.pdf.

Zachary Teed, Lahav Lipson, and Jia Deng. Deep patch visual odometry. In *Advances in Neural Information Processing Systems*, 2023. URL https://papers.nips.cc/paper_files/paper/2023/hash/7ac484b0f1a1719ad5be9aa8c8455fbb-Abstract-Conference.html.

Bill Triggs, Philip McLauchlan, Richard Hartley, and Andrew Fitzgibbon. Bundle adjustment—a modern synthesis. In *Vision Algorithms: Theory and Practice*, volume 1883 of *Lecture Notes in Computer Science*. Springer, 2000. doi: 10.1007/3-540-44480-7_21. URL https://doi.org/10.1007/3-540-44480-7_21.

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. VGGT: Visual geometry grounded transformer. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Wang_VGGT_Visual_Geometry_Grounded_Transformer_CVPR_2025_paper.html.

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. URL https://arxiv.org/abs/2312.14132.

Wenshan Wang, Delong Zhu, Xiangwei Wang, Yaoyu Hu, Yuheng Qiu, Chen Wang, Yafei Hu, Ashish Kapoor, and Sebastian Scherer. TartanAir: A dataset to push the limits of visual SLAM. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020. URL https://arxiv.org/abs/2003.14338.

Yifan Wang, Jianjun Zhou, Haoyi Zhu, Wenzheng Chang, Yang Zhou, Zizun Li, Junyi Chen, Jiangmiao Pang, Chunhua Shen, and Tong He. π^3: Permutation-equivariant visual geometry learning. In *International Conference on Learning Representations (ICLR)*, 2026. URL https://arxiv.org/abs/2507.13347. Pi3X engineering release documented in the official repository, December 28, 2025.

Junyi Zhang, Charles Herrmann, Junhwa Hur, Varun Jampani, Trevor Darrell, Forrester Cole, Deqing Sun, and Ming-Hsuan Yang. MonST3R: A simple approach for estimating geometry in the presence of motion. In *International Conference on Learning Representations (ICLR)*, 2025. URL https://arxiv.org/abs/2410.03825.

A METRIC DEFINITIONS AND AGGREGATION

Throughout evaluation, superscript $*$ denotes ground truth, while (1) and (2) denote the optimization stages in Section 3. Let Ω be the set of view–pixel pairs (i, p) selected by the dataset's valid-target mask, with finite positive target depth $D_i^*(p)$. For predicted depth $D_i(p)$,

$$\mathrm{AbsRel}(D, D^*) = \frac{1}{|\Omega|} \sum_{(i,p)\in\Omega} \frac{|D_i(p) - D_i^*(p)|}{D_i^*(p)}, \qquad s_D = \mathrm{median}_{(i,p)\in\Omega} \frac{D_i^*(p)}{D_i(p)}. \tag{10}$$

Raw depth scoring uses D and aligned scoring uses $s_D D$, with one scale shared by all views and pixels in a window. Experimental reports also retain depth MAE and RMSE.

View 0 defines the reference frame. Camera-only diagnostics form $A_i = R_0^\top R_i$ and $\tau_i = R_0^\top (C_i - C_0)$ in float64, with starred counterparts computed from target poses. Given the singular-value decomposition $A = U_A \Sigma_A V_A^\top$, project by $\mathcal{P}(A) = U_A \,\mathrm{diag}(1, 1, \det(U_A V_A^\top)) V_A^\top$. For $E_i = \mathcal{P}(A_i)^\top \mathcal{P}(A_i^*)$, define $h(E) = (E_{32} - E_{23}, E_{13} - E_{31}, E_{21} - E_{12})^\top$. The rotation error in degrees is

$$R_{\mathrm{err}} = \frac{1}{V-1} \sum_{i=1}^{V-1} \frac{180}{\pi} \,\mathrm{atan2}\left(\frac{\|h(E_i)\|_2}{2}, \frac{\mathrm{tr}\, E_i - 1}{2} \right). \tag{11}$$

Projection is scoring-only and identical for native, BA, and refined cameras. The original rotation diagnostic instead uses $180/\pi$ times $\arccos(\mathrm{clip}((\mathrm{tr}(A_i^\top A_i^*) - 1)/2, -1, 1))$, averaged over the same non-reference views. Translation error averages $\|\tau_i - \tau_i^*\|_2$. Its independently scale-aligned (SA) version uses the nonnegative least-squares scale

$$s_t = \max\left(0, \frac{\sum_{i=1}^{V-1} \tau_i^\top \tau_i^*}{\sum_{i=1}^{V-1} \|\tau_i\|_2^2} \right), \tag{12}$$

when the denominator is positive, then averages $\|s_t \tau_i - \tau_i^*\|_2$ over non-reference views. With a zero denominator, the scorer records the scale as unavailable and evaluates the unchanged zero predicted translations. Depth and translation scales are fitted independently for these diagnostics.

A shared scale for the complete reconstruction. To evaluate depth and motion together, apply the single depth scale s_D to the entire reconstruction. Here relative transforms use the full matrix inverse, $\widetilde{T}_i = T_0^{-1} T_i$, with blocks $\widetilde{R}_i, \widetilde{t}_i$ and analogous target quantities. This follows the reconstruction scorer's arithmetic; camera-only diagnostics above retain their transpose-based convention. For $\bar{p} = (p_x, p_y, 1)^\top$, the predicted and target points in their respective first-camera frames are

$$X_i(p) = \widetilde{R}_i D_i(p) K_i^{-1} \bar{p} + \widetilde{t}_i, \qquad X_i^*(p) = \widetilde{R}_i^* D_i^*(p) (K_i^*)^{-1} \bar{p} + \widetilde{t}_i^*. \tag{13}$$

Here K_i is the fixed inference intrinsic matrix and K_i^* is target calibration, both mapped to the actual scoring pixel grid. Known-K experiments use $K_i = K_i^*$; Predicted-K experiments use native K_i for prediction rays and K_i^* for target rays. Target calibration never enters Predicted-K inference or optimization. The primary point-relative error P and shared-scale translation error ST are

$$P = \frac{1}{|\Omega|} \sum_{(i,p)\in\Omega} \frac{\|s_D X_i(p) - X_i^*(p)\|_2}{D_i^*(p)}, \qquad \mathrm{ST} = \frac{1}{V-1} \sum_{i=1}^{V-1} \|s_D \widetilde{t}_i - \widetilde{t}_i^*\|_2. \tag{14}$$

P is dimensionless; ST is computed in metres and multiplied by 100 for table values in centimetres. The same scalar multiplies point coordinates and camera translations. Reports additionally retain point MAE (the P numerator averaged without depth normalization). They also retain $|\log(s_D / s_t^{\mathrm{rec}})|$ for positive s_t^{rec}, obtained from Equation (12) using $\widetilde{t}_i, \widetilde{t}_i^*$ in place of τ_i, τ_i^*. NuScenes point targets cover the reference view, while its shared-scale camera score uses the complete pose window. TartanAir and TUM point scores combine available depths and poses across views.

All scoring uses float64 geometry. Requested, valid, and scored support counts are retained. Confidence values do not change evaluation support. For multiple recording groups, we compute the arithmetic mean of window metrics within each group, then the arithmetic mean of those group means. Window-level paired distributions separately report mean and median change, 95th-percentile change, and fraction with more than 5% relative worsening. Every prescribed window is included; undefined relative changes at zero native error are counted explicitly.

B IMPLEMENTATION AND REPRODUCIBILITY

Native models and preprocessing. We use frozen public MapAnything and Pi3X models, with source revisions `3d10cf7` and `9fa3ddb`. MapAnything uses the `facebook/map-anything` checkpoint at revision `a1d87e9`; Pi3X uses the released `yyfz233/Pi3X` model. Both process all views together, in evaluation mode with parameter gradients disabled. MapAnything receives calibrated rays and normalized RGB (channel means $0.485, 0.456, 0.406$ and standard deviations $0.229, 0.224, 0.225$), with depth/pose conditioning disabled. Its forward pass uses float32 with TF32 enabled. Pi3X receives RGB in $[0, 1]$, normalizes internally, and uses bfloat16 autocast with TF32 disabled. Both export camera-z depth and camera-to-world poses. Supplementary metadata gives full source identifiers and checkpoint provenance.

NuScenes combines front (F) and front-left (L) cameras at key-frame offsets $(0, 2, 4)$ in the exact order $(F_4, F_0, L_0, F_2, L_2, L_4)$, with F_4 as reference. We bilinearly resize complete RGB images to 336×560, without cropping, and adjust intrinsics by the half-pixel convention on each axis. Depth targets use the reference sample's LiDAR sweep, retaining sensor ranges 2–60 m and projected camera depths $0.1 < z < 80$ m. Projection retains the nearest return per reference-view pixel. TartanAir uses nonoverlapping consecutive three-frame windows from the Hard left-camera sequences, resized from 480×640 to 336×448. Intrinsics become $(f_x, f_y, c_x, c_y) = (224, 224, 223.85, 167.85)$. Depth is nearest-neighbor resampled with support $0 < z < 80$ m; camera axes are converted from NED to optical while retaining the original world frame. No target depth, pose, or scale enters native inference or refinement. Exact evaluation rosters accompany the code.

Persistent and temporary states. Each source-view/integer-pixel key has one nuisance scalar shared across its target matches. Component centering weights these unique keys equally; components without matched source rays have no nuisance variables. Stage 2 starts from the Stage 1 coarse coefficients and retains the original native depth as its multiplicative reference. It reuses the exact saved cameras throughout dense refitting. The algebraic centering identities hold up to floating-point rounding in numerical evaluation.

The grid has $\lfloor H/14 \rfloor \lfloor W/14 \rfloor$ coefficients per view. Camera scale m uses the float32 tensor median, whereas the center-prior scale uses the float64 median with even-length averaging. Stage 1 raw variables start at zero and receive 320 Adam updates; Stage 2 copies the fitted depth coefficients and restarts Adam for 80 depth-only updates, both at learning rate 0.03. Computation uses deterministic CUDA settings and seed 42 where an API exposes a seed. The method has no learned refiner parameters or training phase.

Persistence and scoring. Compact outputs store the final coarse field, exact cameras, method identity, status, and source role. They are replayed before supervision is loaded. Windows with fewer than 16 retained directed matches return the native prediction before Stage 1. A non-finite loss, gradient, or parameter state in either stage also triggers a recorded native fallback; Stage 2 follows a completed Stage 1. These checks use only inference inputs. The 2400 backbone–window evaluations have no recorded native fallback in the tested refinement/control arms. Every prescribed input is retained, including disconnected match graphs. Code, manifests, and metric evidence are separated from third-party datasets and pretrained weights; cached reproduction does not require backbone retraining.

Control definitions. The joint control is the persistent Stage 1 depth and cameras, without emitting nuisance. Candidate cameras with native depth use unmodified native depth. The candidate cold-refit control starts the same fixed-camera 80-step refit at zero depth coefficients. Ours uses the Stage 1 coefficients to initialize that refit. Corresponding BA arms use exact exported BA cameras with native depth or the same zero-initialized refit. The factorial's soft arm decodes $\ell_s = 0.7 \tanh \eta_s$ with penalty $10^{-6}\langle \ell^2 \rangle_{\mathcal{U}}$, while the centered arm uses Equation (4). All other factors are fixed within this 2×2 experiment.

C DATA COVERAGE AND FIXED-METHOD RECORDING PROTOCOL

The 600 distinct windows comprise 33 nuScenes windows in ten recording logs, 320 TartanAir windows in three environments, and 247 TUM windows in five recordings: 18 physical groups

Table 6: Recording-level results show the contributions to the equal-recording fresh TUM summary. Shared-scale point error is relative; shared-scale translation T is in cm. N counts windows. Best displayed values and ties are bold.

Recording	Model	N	Shared point relative ↓			Shared T (cm) ↓		
			Native	BA+refit	Ours	Native	BA+refit	Ours
Far	MapAnything	15	0.0872	0.0878	**0.0842**	5.64	**3.77**	3.87
Teddy	MapAnything	77	**0.1784**	0.1926	0.1786	3.90	3.41	**3.04**
Far	Pi3X	15	0.0245	0.0280	**0.0227**	4.44	3.83	**2.82**
Teddy	Pi3X	77	0.1115	0.1149	**0.1108**	1.67	1.79	**1.64**

in total. TartanAir DEV uses `forestwinter`; CAL uses `soulcity` and `westerndesert`. DEV contributes 274 windows, previously evaluated cohorts contribute 234, and the new recordings contribute 92. All four backbones process the same windows: 2400 model–window evaluations, not 2400 independent inputs.

Historical TUM recordings are `long_office_household` (83 windows), `structure_texture_near` (35), and `cabinet` (37). New recordings are `nostructure_texture_far` (15) and `teddy` (77). All are Freiburg-3 recordings from the official TUM RGB-D distribution (Sturm et al., 2012). RGB, registered depth, and poses are greedily associated one-to-one by timestamp with maximum offset $0.03\,\mathrm{s}$. Windows use associated-frame offsets $(0, 10, 20)$, starts advance by 30 associated frames, and every complete window is used. The two new recording identities, sampling rule, methods, and optimization settings were fixed before their backbone inference.

Freiburg-3 images are undistorted. Source calibration is $(f_x, f_y, c_x, c_y) = (535.4, 539.2, 320.1, 247.6)$. RGB is resized from 640×480 to 448×336 with bilinear interpolation, registered projective depth with nearest-neighbor interpolation. With resize factor $\alpha = 0.7$, intrinsics follow $f_x' = \alpha f_x$, $f_y' = \alpha f_y$, $c_x' = \alpha(c_x + 0.5) - 0.5$, and $c_y' = \alpha(c_y + 0.5) - 0.5$. Here f_x, f_y are pixel focal lengths and (c_x, c_y) is the principal point; primes denote resized values. Integer depth is divided by 5000 to obtain metres; zero is invalid. Target camera-to-world transforms are float64. Targets define evaluation support, never predicted corrections.

The new recordings test transfer of the locked refiner to additional real scenes from the same sensor family as historical TUM. Their fresh status concerns refiner development; backbone pretraining membership is unspecified.

D GEOMETRIC BASELINES

Native-prior bundle adjustment. We use pycolmap 4.0.4 with Ceres. Each directed RGB match creates an independent two-observation landmark; reverse-direction matches remain separate. Landmarks initialize by backprojecting native source camera-z depth through the protocol's fixed intrinsics. Each view has a fixed PINHOLE camera with its own f_x, f_y, c_x, c_y: Known K or the same Predicted K used by NativeRefine. No focal-length or principal-point refinement is enabled. Native float32 rotations are projected to $\mathrm{SO}(3)$ while preserving camera centers before constructing the rigid-pose problem.

BA optimizes landmarks and cameras with quadratic native-center priors of standard deviation $0.02\,\mathrm{median}_{64}(D^0)$. The complete first pose of each connected match component is fixed; isolated cameras remain native. The solver uses unweighted two-dimensional block soft-L_1 loss with scale $0.005\max(H, W)$, at most 100 Ceres iterations, and four CPU threads. Rotation and landmark depth are controlled through reprojection, with no direct priors. Both methods share measurements and native-center prior scale; NativeRefine uses the overlap-weighted coordinate Huber objective in Equation (6).

For dense evaluation, exact exported BA cameras are paired with native depth or the native-start 80-step dense refit, using the same numerical routine as the candidate-camera cold-refit control. Dense scoring uses these full depth maps, with GT alignment applied only during evaluation. All windows produce scoreable BA outputs; solver convergence and exhaustion of the iteration budget remain separately recorded, without filtering by termination status.

Table 7: NativeRefine achieves lower camera errors than official EPO in all ten previously evaluated cohort–model cells. Each cell is original acos rotation (degrees) / independently scale-aligned (SA) translation (cm). The camera-only comparison shares native inputs and complete coverage; EPO uses edge alignment, focal/depth optimization and a variable iteration budget. Matched dense-refit results and the uniform SO(3) diagnostic appear in Tables 2 and 3.

			Original R (°) ↓ / SA T (cm) ↓			
Observed cohort	Model	N/G	Native	BA-prior	EPO	Ours
nuScenes DEV	Map	18/6	1.405 / 40.128	1.147 / 26.309	0.946 / 29.778	0.917 / 28.868
nuScenes DEV	π^3X	18/6	1.107 / 30.133	1.080 / 20.147	3.235 / 49.650	0.780 / 21.820
nuScenes CAL	Map	15/4	1.196 / 22.162	1.241 / 24.093	1.349 / 31.616	0.790 / 18.763
nuScenes CAL	π^3X	15/4	0.667 / 14.316	1.349 / 16.750	1.580 / 19.902	0.620 / 12.587
TartanAir DEV	Map	256/1	0.456 / 3.775	0.022 / 0.831	0.371 / 1.294	0.134 / 0.653
TartanAir DEV	π^3X	256/1	0.150 / 1.487	0.021 / 0.392	0.100 / 0.619	0.068 / 0.317
TartanAir CAL	Map	64/2	0.496 / 5.070	0.118 / 1.934	1.111 / 6.283	0.176 / 1.437
TartanAir CAL	π^3X	64/2	0.161 / 2.170	0.120 / 1.374	0.923 / 3.213	0.146 / 1.155
TUM observed	Map	155/3	0.736 / 1.278	0.537 / 0.874	0.762 / 1.957	0.562 / 0.754
TUM observed	π^3X	155/3	0.384 / 0.643	0.538 / 0.750	0.480 / 0.858	0.411 / 0.655

Table 8: nuScenes DEV: depth, camera, and shared-scale errors resolve the effects of the seven method variants. D denotes depth, R rotation, T translation, and P point error; SA means independently scale-aligned, while shared P and T use one common scale. Results are equal-group means; best displayed values and ties are bold. Joint SO(3) diagnostic unavailable (–).

Model	Method	Raw D AbsRel ↓	SA D AbsRel ↓	Orig. R (°) ↓	SO(3) R (°) ↓	SA T (cm) ↓	Shared P relative ↓	Shared T (cm) ↓
MapAnything	Native	0.1444	0.0836	1.405	1.405	40.13	0.0903	53.10
MapAnything	Centered joint	**0.1443**	0.0835	**0.917**	–	28.87	0.0902	43.48
MapAnything	NativeRefine	0.1466	0.0812	**0.917**	**0.917**	28.87	0.0876	44.76
MapAnything	BA + native depth	0.1444	0.0836	1.147	1.147	**26.31**	0.0903	42.83
MapAnything	BA + refit	0.1446	**0.0809**	1.147	1.147	**26.31**	**0.0873**	**41.33**
MapAnything	Ours cameras + native depth	0.1444	0.0836	**0.917**	**0.917**	28.87	0.0903	43.53
MapAnything	Ours cameras + cold refit	0.1466	0.0812	**0.917**	**0.917**	28.87	0.0876	44.76
Pi3X	Native	**0.3541**	0.0579	1.107	1.111	30.13	0.0623	35.81
Pi3X	Centered joint	**0.3541**	0.0580	**0.780**	–	21.82	0.0624	29.38
Pi3X	NativeRefine	0.3556	0.0566	**0.780**	**0.784**	21.82	0.0608	30.63
Pi3X	BA + native depth	**0.3541**	0.0579	1.080	1.080	**20.15**	0.0623	**28.17**
Pi3X	BA + refit	0.3545	**0.0563**	1.080	1.080	**20.15**	**0.0606**	28.31
Pi3X	Ours cameras + native depth	**0.3541**	0.0579	**0.780**	**0.784**	21.82	0.0623	29.46
Pi3X	Ours cameras + cold refit	0.3555	0.0565	**0.780**	**0.784**	21.82	0.0608	30.63

Official EPO on historical cohorts. The supplemental camera comparison uses the unmodified official EPO implementation (D'Urso et al., 2026), revision `e3367186`. Its `EPO.from_ff` interface receives frozen native RGB, depths, poses, and intrinsics, with the Triton backend, sequential window five, seed 42, focal/depth optimization and pose-MLP updates enabled. It uses the released sparse-window pose convergence rule, at most 2000 iterations, default Canny processing, and a separate camera per image. Ground truth never enters optimization.

EPO initializes SIMPLE_PINHOLE focal length by averaging f_x and f_y, then refines it; NativeRefine and BA retain exact fixed PINHOLE calibration. We add 0.5 to integer-centered principal points at the interface because `from_ff` internally converts COLMAP half-integer centers. Camera-to-world matrices are inverted for its API and exported poses are inverted back for scoring. This camera comparison covers the ten historical cells and applies the same original $\arccos$ rotation/SA-translation scorer to every method. Main camera results use the uniform $\mathrm{SO}(3)$ diagnostic.

E Complete reconstruction controls

All seven delivered reconstructions use common evaluation support and equal physical-group weights. Raw and aligned depth, as well as original and uniformly projected rotation, resolve complementary aspects of the output. The independently exported projected-rotation diagnostic is unavailable for DEV joint and is marked "–"; its original rotation and reconstruction scores are reported.

Table 9: TartanAir DEV: depth, camera, and shared-scale errors resolve the effects of the seven method variants. D denotes depth, R rotation, T translation, and P point error; SA means independently scale-aligned, while shared P and T use one common scale. Results are equal-group means; best displayed values and ties are bold. Joint SO(3) diagnostic unavailable (–).

Model	Method	Raw D AbsRel ↓	SA D AbsRel ↓	Orig. R (°) ↓	SO(3) R (°) ↓	SA T (cm) ↓	Shared P relative ↓	Shared T (cm) ↓
MapAnything	Native	1.5223	0.2227	0.456	0.456	3.77	0.2761	5.59
MapAnything	Centered joint	1.5223	0.2227	0.134	–	**0.65**	0.2746	1.98
MapAnything	NativeRefine	**1.5212**	**0.2208**	0.134	0.137	**0.65**	**0.2721**	**1.92**
MapAnything	BA + native depth	1.5223	0.2227	**0.022**	**0.022**	0.83	0.2753	3.50
MapAnything	BA + refit	1.5224	0.2216	**0.022**	**0.022**	0.83	0.2738	3.41
MapAnything	Ours cameras + native depth	1.5223	0.2227	0.134	0.137	**0.65**	0.2746	1.98
MapAnything	Ours cameras + cold refit	**1.5212**	**0.2208**	0.134	0.137	**0.65**	**0.2721**	**1.92**
Pi3X	Native	0.1705	0.1414	0.150	0.158	1.49	0.1750	2.55
Pi3X	Centered joint	0.1705	0.1414	0.068	–	**0.32**	0.1742	1.46
Pi3X	NativeRefine	**0.1701**	**0.1411**	0.068	0.064	**0.32**	**0.1737**	**1.40**
Pi3X	BA + native depth	0.1705	0.1414	**0.021**	**0.021**	0.39	0.1748	1.95
Pi3X	BA + refit	**0.1701**	0.1417	**0.021**	**0.021**	0.39	0.1750	1.85
Pi3X	Ours cameras + native depth	0.1705	0.1414	0.068	0.064	**0.32**	0.1742	1.46
Pi3X	Ours cameras + cold refit	**0.1701**	**0.1411**	0.068	0.064	**0.32**	**0.1737**	**1.40**

Table 10: nuScenes CAL: depth, camera, and shared-scale errors resolve the effects of the seven method variants. D denotes depth, R rotation, T translation, and P point error; SA means independently scale-aligned, while shared P and T use one common scale. Results are equal-group means; best displayed values and ties are bold.

Model	Method	Raw D AbsRel ↓	SA D AbsRel ↓	Orig. R (°) ↓	SO(3) R (°) ↓	SA T (cm) ↓	Shared P relative ↓	Shared T (cm) ↓
MapAnything	Native	**0.1974**	0.1546	1.196	1.196	22.16	0.1668	28.73
MapAnything	Centered joint	**0.1974**	0.1546	**0.790**	**0.791**	**18.76**	0.1668	26.71
MapAnything	NativeRefine	0.1988	**0.1537**	**0.790**	**0.791**	**18.76**	**0.1657**	26.96
MapAnything	BA + native depth	**0.1974**	0.1546	1.241	1.241	24.09	0.1668	30.82
MapAnything	BA + refit	0.1988	0.1539	1.241	1.241	24.09	0.1660	30.83
MapAnything	Ours cameras + native depth	**0.1974**	0.1546	**0.790**	**0.791**	**18.76**	0.1668	**26.70**
MapAnything	Ours cameras + cold refit	0.1988	**0.1537**	**0.790**	**0.791**	**18.76**	0.1658	26.95
Pi3X	Native	**0.3795**	**0.0841**	0.667	0.668	14.32	**0.0898**	17.19
Pi3X	Centered joint	0.3797	**0.0841**	**0.620**	**0.619**	**12.59**	**0.0898**	15.61
Pi3X	NativeRefine	0.3804	**0.0841**	**0.620**	**0.619**	**12.59**	**0.0898**	15.85
Pi3X	BA + native depth	**0.3795**	**0.0841**	1.349	1.349	16.75	**0.0898**	20.08
Pi3X	BA + refit	0.3811	0.0842	1.349	1.349	16.75	0.0900	20.13
Pi3X	Ours cameras + native depth	**0.3795**	**0.0841**	**0.620**	**0.619**	**12.59**	**0.0898**	**15.49**
Pi3X	Ours cameras + cold refit	0.3804	**0.0841**	**0.620**	**0.619**	**12.59**	0.0899	15.85

F PAIRED-WINDOW DISTRIBUTIONS AND COMPUTATION

Table 14 reports window-weighted paired deltas over all windows, complementing group-balanced cohort means. Relative worsening uses positive reference denominators; absolute deltas include zero-reference windows as well. Main-text percentage gains are computed from ratios of group-balanced means.

Table 15 reports per-window A6000 timings. Cached-input rows measure file reading; fresh rows measure native and matcher forwards. Joint and refit timings cover optimization, with scoring, artifact I/O, and network initialization excluded. Measurements use the stated backbone precision settings under concurrent cluster execution.

Table 11: TartanAir CAL: depth, camera, and shared-scale errors resolve the effects of the seven method variants. D denotes depth, R rotation, T translation, and P point error; SA means independently scale-aligned, while shared P and T use one common scale. Results are equal-group means; best displayed values and ties are bold.

Model	Method	Raw D AbsRel ↓	SA D AbsRel ↓	Orig. R (°) ↓	SO(3) R (°) ↓	SA T (cm) ↓	Shared P relative ↓	Shared T (cm) ↓
MapAnything	Native	0.3417	0.1405	0.496	0.496	5.07	0.1791	8.54
MapAnything	Centered joint	0.3417	0.1405	0.176	0.179	**1.44**	0.1773	3.36
MapAnything	NativeRefine	**0.3403**	**0.1381**	0.176	0.179	**1.44**	**0.1742**	**3.27**
MapAnything	BA + native depth	0.3417	0.1405	**0.118**	**0.118**	1.93	0.1786	5.88
MapAnything	BA + refit	0.3415	0.1395	**0.118**	**0.118**	1.93	0.1773	5.81
MapAnything	Ours cameras + native depth	0.3417	0.1405	0.176	0.179	**1.44**	0.1773	3.36
MapAnything	Ours cameras + cold refit	**0.3403**	**0.1381**	0.176	0.179	**1.44**	**0.1742**	**3.27**
Pi3X	Native	0.1170	**0.0540**	0.161	0.169	2.17	0.0707	3.12
Pi3X	Centered joint	0.1170	0.0541	0.146	0.150	**1.16**	0.0705	2.86
Pi3X	NativeRefine	**0.1167**	**0.0540**	0.146	0.150	**1.16**	**0.0700**	**2.78**
Pi3X	BA + native depth	0.1170	**0.0540**	**0.120**	**0.120**	1.37	0.0704	2.85
Pi3X	BA + refit	0.1169	0.0546	**0.120**	**0.120**	1.37	0.0706	**2.78**
Pi3X	Ours cameras + native depth	0.1170	**0.0540**	0.146	0.150	**1.16**	0.0705	2.86
Pi3X	Ours cameras + cold refit	**0.1167**	**0.0540**	0.146	0.150	**1.16**	**0.0700**	**2.78**

Table 12: TUM observed: depth, camera, and shared-scale errors resolve the effects of the seven method variants. D denotes depth, R rotation, T translation, and P point error; SA means independently scale-aligned, while shared P and T use one common scale. Results are equal-group means; best displayed values and ties are bold.

Model	Method	Raw D AbsRel ↓	SA D AbsRel ↓	Orig. R (°) ↓	SO(3) R (°) ↓	SA T (cm) ↓	Shared P relative ↓	Shared T (cm) ↓
MapAnything	Native	**0.7494**	0.0393	0.736	0.736	1.28	0.0445	1.91
MapAnything	Centered joint	**0.7494**	0.0393	0.562	0.562	**0.75**	0.0437	**1.34**
MapAnything	NativeRefine	0.7495	**0.0389**	0.562	0.562	**0.75**	**0.0433**	**1.34**
MapAnything	BA + native depth	**0.7494**	0.0393	**0.537**	**0.537**	0.87	0.0457	1.57
MapAnything	BA + refit	0.7504	0.0415	**0.537**	**0.537**	0.87	0.0478	1.58
MapAnything	Ours cameras + native depth	**0.7494**	0.0393	0.562	0.562	**0.75**	0.0437	**1.34**
MapAnything	Ours cameras + cold refit	0.7495	**0.0389**	0.562	0.562	**0.75**	**0.0433**	**1.34**
Pi3X	Native	**0.3013**	0.0253	**0.384**	**0.387**	**0.64**	0.0289	**0.77**
Pi3X	Centered joint	**0.3013**	0.0253	0.411	0.414	0.66	0.0290	0.80
Pi3X	NativeRefine	**0.3013**	**0.0251**	0.411	0.414	0.66	**0.0288**	0.80
Pi3X	BA + native depth	**0.3013**	0.0253	0.538	0.538	0.75	0.0302	0.88
Pi3X	BA + refit	0.3015	0.0267	0.538	0.538	0.75	0.0315	0.88
Pi3X	Ours cameras + native depth	**0.3013**	0.0253	0.411	0.414	0.66	0.0290	0.80
Pi3X	Ours cameras + cold refit	**0.3013**	**0.0251**	0.411	0.414	0.66	**0.0288**	0.80

Table 13: TUM fresh: depth, camera, and shared-scale errors resolve the effects of the seven method variants. D denotes depth, R rotation, T translation, and P point error; SA means independently scale-aligned, while shared P and T use one common scale. Results are equal-group means; best displayed values and ties are bold.

Model	Method	Raw D AbsRel ↓	SA D AbsRel ↓	Orig. R (°) ↓	SO(3) R (°) ↓	SA T (cm) ↓	Shared P relative ↓	Shared T (cm) ↓
MapAnything	Native	**0.5033**	0.1218	1.345	1.345	3.24	0.1328	4.77
MapAnything	Centered joint	**0.5033**	0.1219	1.042	1.042	1.87	0.1318	**3.45**
MapAnything	NativeRefine	0.5034	**0.1215**	1.042	1.042	1.87	**0.1314**	**3.45**
MapAnything	BA + native depth	**0.5033**	0.1218	**0.721**	**0.721**	**1.55**	0.1341	3.52
MapAnything	BA + refit	0.5054	0.1280	**0.721**	**0.721**	**1.55**	0.1402	3.59
MapAnything	Ours cameras + native depth	**0.5033**	0.1218	1.042	1.042	1.87	0.1318	**3.45**
MapAnything	Ours cameras + cold refit	0.5034	**0.1215**	1.042	1.042	1.87	**0.1314**	**3.45**
Pi3X	Native	0.3572	0.0618	**0.614**	**0.616**	1.82	0.0680	3.05
Pi3X	Centered joint	0.3572	0.0618	0.664	0.666	**1.33**	0.0674	2.24
Pi3X	NativeRefine	**0.3567**	**0.0612**	0.664	0.666	**1.33**	**0.0667**	**2.23**
Pi3X	BA + native depth	0.3572	0.0618	0.724	0.724	1.44	0.0697	2.79
Pi3X	BA + refit	0.3569	0.0638	0.724	0.724	1.44	0.0715	2.81
Pi3X	Ours cameras + native depth	0.3572	0.0618	0.664	0.666	**1.33**	0.0674	2.24
Pi3X	Ours cameras + cold refit	**0.3567**	**0.0612**	0.664	0.666	**1.33**	**0.0667**	**2.23**

Table 14: Paired-window deltas quantify typical gains and upper-tail worsening relative to native. Deltas are NativeRefine minus native: mean, median, and 95th percentile. Negative deltas indicate improvement; lower is better for every reported statistic. W counts windows worsening by more than 5% over nonzero-reference pairs; all windows contribute to delta statistics. Shared-scale point deltas are relative and translation deltas are cm.

Cohort	Model	Shared-scale point delta (relative)				Shared-scale T delta (cm)			
		Mean ↓	Median ↓	p95 ↓	W ↓	Mean ↓	Median ↓	p95 ↓	W ↓
nuScenes DEV	MapAnything	-0.0027	-0.0024	0.0009	1/18	-8.34	-7.32	-0.57	0/18
nuScenes DEV	Pi3X	-0.0015	-0.0017	0.0009	1/18	-5.19	-4.55	-1.68	0/18
TartanAir DEV	MapAnything	-0.0039	-0.0029	0.0004	0/256	-3.67	-3.28	-0.73	0/256
TartanAir DEV	Pi3X	-0.0013	-0.0008	0.0014	1/256	-1.16	-1.38	3.16	33/256
nuScenes CAL	MapAnything	-0.0013	0.0000	0.0011	0/15	-2.06	-4.04	5.60	4/15
nuScenes CAL	Pi3X	0.0000	0.0000	0.0029	1/15	-0.91	-1.47	7.79	5/15
TartanAir CAL	MapAnything	-0.0049	-0.0020	-0.0003	0/64	-5.27	-3.43	-0.90	1/64
TartanAir CAL	Pi3X	-0.0006	-0.0004	0.0058	3/64	-0.34	-1.19	7.49	9/64
TUM observed	MapAnything	-0.0014	-0.0012	0.0016	4/155	-0.63	-0.59	0.16	12/155
TUM observed	Pi3X	-0.0003	-0.0002	0.0019	14/155	0.03	0.01	0.76	71/155
TUM fresh	MapAnything	-0.0003	-0.0003	0.0100	11/92	-1.01	-0.81	0.36	7/92
TUM fresh	Pi3X	-0.0009	-0.0003	0.0041	6/92	-0.29	-0.06	0.83	32/92

Table 15: Joint optimization accounts for most NativeRefine refinement time. Values are measured seconds per window across final-evaluation cases. Native and matcher columns report cache I/O in cache rows and live inference in forward rows. Joint/refit/total cover candidate optimization; times use window-weighted averages. Model initialization is recorded separately in whole-job time.

Cohort	Model	Input mode	Native (s) ↓	Matcher (s) ↓	Joint (s) ↓	Refit (s) ↓	Total (s) ↓
nuScenes CAL	MapAnything	cache	0.080	0.016	2.596	0.351	2.947
nuScenes CAL	Pi3X	cache	0.069	0.020	4.141	0.537	4.678
TartanAir CAL	MapAnything	cache	0.089	0.013	2.489	0.309	2.798
TartanAir CAL	Pi3X	cache	0.076	0.008	4.033	0.527	4.560
TUM observed	MapAnything	cache	0.099	0.015	2.451	0.310	2.761
TUM observed	Pi3X	cache	0.076	0.014	3.916	0.513	4.429
TUM fresh	MapAnything	forward	0.264	0.624	2.381	0.307	2.688
TUM fresh	Pi3X	forward	0.287	0.655	3.971	0.529	4.499

G RGB-ONLY TRANSFER WITH FIXED PREDICTED INTRINSICS

Frozen native models and pixel grids. VGGT uses the public VGGT-1B checkpoint, revision `860abec7937d`, with source `a288dd0f1478`. DA3 uses DA3-GIANT-1.1, checkpoint revision `72ee9f89ce4e`, with source `3d835ec1a580`. Both pose and intrinsic conditioning are disabled in DA3 (`extrinsics=None`, `intrinsics=None`); ray-pose decoding is enabled, the reference strategy is `saddle_balanced`, and Gaussian output is disabled.

Both adapters convert OpenCV world-to-camera poses to camera-to-world exactly once. Official VGGT 518-pixel and DA3 504-pixel upper-bound preprocessing operate on the stored evaluation RGB images. Depth and predicted K are coherently mapped back to the declared evaluation grid, where matches are defined: 336×560 for nuScenes and 336×448 for TUM/TartanAir. Only RGB enters the backbones; their predicted depth, cameras, and K initialize refinement. No ground-truth pose, depth, or external calibration enters inference. Table 1 separates these inputs from the calibrated experiments.

Matched controls and scoring. The native, BA+refit, no-nuisance+refit, and full NativeRefine arms share each forward pass and its fixed K. The no-nuisance arm holds $\ell = 0$ throughout Stage 1 and excludes its variables from the optimizer; its own depth coefficients and cameras initialize the subsequent 80-step fixed-camera refit, with fresh Adam state and the original native depth basis. The full method uses exactly the original 320+80 schedule, bounds, losses, and weights. No backbone-specific search is performed. Tables 16 and 17 retain all four arms, roles, and point/translation/rotation outcomes. Appendix I adds the corresponding strict no-nuisance controls for MapAnything and Pi3X, complementing the centered-versus-uncentered factorial in Table 5.

The dual-intrinsics scorer in Equation (13) separates prediction rays from target rays. Regression checks cover equality with the calibrated scorer when the two K matrices agree, nonzero 3D error from an incorrect predicted focal length even at correct depth/pose, and invariance to a common positive scaling of predicted depths and camera translations. The same single depth-derived scale is used throughout; no second alignment removes predicted calibration error.

Transfer behavior. Across the twelve Predicted-K cohort–backbone cells, point error improves over native in nine and over BA+refit in ten. Shared-scale translation improves over native in three, and rotation in two. The complete method improves both primary errors over the strict no-nuisance+refit control in five cells, and rotation in eight. On the equal-recording fresh TUM mean, it lowers point/translation/rotation errors relative to that control by $0.41/5.20/2.43\%$ for VGGT and $0.59/7.40/4.38\%$ for DA3. Relative to native, fresh TUM point error decreases $0.35/0.73\%$, translation changes by $-2.34/+1.69\%$, and rotation increases $13.92/11.45\%$. These results establish executable cross-backbone transfer and useful dense corrections; stable camera improvement remains concentrated in the calibrated experiments. Because both backbone and K source change between protocols, these differences do not isolate calibration as the cause.

H CONSUMER-GPU EXECUTION

All 1200 VGGT/DA3 model–window evaluations completed on one RTX 5070 Ti with 16 GB of VRAM, using native inference, RGB matching, and refinement in separate processes. The native model and matcher weights are released between stages; they need not reside on the GPU together. DA3 reuses the same frozen RGB matches computed for VGGT. This is a practical memory scheduling property of the pipeline, not a smaller foundation model.

Table 18 separates measured native inference from cached-input loading and separates network stages from optimization. Cold calls include model initialization; warm means exclude the first native call in each job. Reported peaks are PyTorch tensor allocations for native inference and matching, not whole-process or refiner-only VRAM. Refiner timings cover the full 320+80 updates. Scoring, artifact I/O, and process setup are excluded from these stage timings, so their sum is not a measured end-to-end latency. The A6000 and 5070 Ti experiments use different backbones and are not a controlled GPU speed comparison.

Table 16: VGGT fixed-predicted-K transfer on all 600 frozen windows. The matched no-nuisance arm fixes $\ell = 0$ during Stage 1 and uses its own warm-start Stage 2. All four arms share native predictions, K and RGB matches within each window. P is shared-scale point error; ST uses the same fitted scale; R is rotation. Equal physical-group means; bold marks displayed minima and ties.

Cohort	Method	P relative ↓	ST (cm) ↓	R (°) ↓
nuScenes DEV	Native	0.1075	49.53	1.509
	BA + refit	0.1052	49.27	2.721
	No nuisance + refit	0.1053	50.91	1.538
	NativeRefine	**0.1047**	**46.88**	**1.491**
TartanAir DEV	Native	**0.3116**	**8.23**	**0.357**
	BA + refit	0.3117	8.33	0.462
	No nuisance + refit	0.3117	8.47	0.404
	NativeRefine	0.3119	8.79	0.422
nuScenes CAL	Native	0.1475	19.38	**1.538**
	BA + refit	0.1475	38.23	2.634
	No nuisance + refit	**0.1472**	20.43	1.636
	NativeRefine	**0.1472**	**19.04**	1.617
TartanAir CAL	Native	0.2191	9.62	**0.339**
	BA + refit	0.2194	**9.54**	0.494
	No nuisance + refit	0.2194	9.75	0.407
	NativeRefine	**0.2190**	10.65	0.423
TUM observed	Native	**0.0869**	**1.16**	**0.429**
	BA + refit	0.0878	1.26	0.565
	No nuisance + refit	0.0870	1.40	0.547
	NativeRefine	0.0871	1.26	0.480
TUM fresh	Native	0.1665	6.64	**0.797**
	BA + refit	0.1699	6.73	1.013
	No nuisance + refit	0.1666	6.84	0.931
	NativeRefine	**0.1659**	**6.48**	0.908

Table 17: DA3 fixed-predicted-K transfer on all 600 frozen windows. The matched no-nuisance arm fixes $\ell = 0$ during Stage 1 and uses its own warm-start Stage 2. All four arms share native predictions, K and RGB matches within each window. P is shared-scale point error; ST uses the same fitted scale; R is rotation. Equal physical-group means; bold marks displayed minima and ties.

Cohort	Method	P relative ↓	ST (cm) ↓	R (°) ↓
nuScenes DEV	Native	0.1071	**39.25**	**2.611**
	BA + refit	0.1045	45.13	4.145
	No nuisance + refit	0.1048	40.83	2.753
	NativeRefine	**0.1042**	40.48	2.783
TartanAir DEV	Native	0.2190	0.59	0.045
	BA + refit	0.2189	**0.46**	**0.036**
	No nuisance + refit	0.2188	0.72	0.070
	NativeRefine	**0.2186**	0.95	0.044
nuScenes CAL	Native	0.1443	24.15	**2.710**
	BA + refit	0.1432	25.05	3.328
	No nuisance + refit	**0.1430**	**23.40**	3.418
	NativeRefine	0.1432	24.55	3.144
TartanAir CAL	Native	0.1066	**1.02**	**0.060**
	BA + refit	0.1070	1.46	0.118
	No nuisance + refit	0.1063	2.16	0.115
	NativeRefine	**0.1058**	2.15	0.128
TUM observed	Native	**0.0409**	**0.70**	**0.379**
	BA + refit	0.0428	0.86	0.546
	No nuisance + refit	0.0410	0.81	0.427
	NativeRefine	0.0410	0.86	0.417
TUM fresh	Native	0.1301	**3.54**	**0.596**
	BA + refit	0.1315	3.60	0.743
	No nuisance + refit	0.1299	3.89	0.694
	NativeRefine	**0.1292**	3.60	0.664

Table 18: Local stage costs on one RTX 5070 Ti (16 GB). Cold native is the single first call including model load; warm native averages the other windows. Matching and full 320+80 refinement are window-mean seconds. †: DA3 reuses the corresponding measured RGB-match cache, not a second matching run. Peaks are allocated tensor MiB for native/matching; refiner peak was not recorded. These separate stages exclude outer export/scoring and are not measured end-to-end latency.

Cohort (n)	Model	Time per window (s)				Peak (MiB)	
		Native cold	Native warm	Matching	Refiner	Native	Matcher
nuScenes (33)	VGGT	58.816	0.770	3.993	10.172	8429	3639
TartanAir (320)	VGGT	20.671	0.320	0.657	7.271	7121	4286
TUM observed (155)	VGGT	15.604	0.300	0.639	6.639	7121	4286
TUM far (15)	VGGT	23.213	0.349	0.746	7.244	7121	4286
TUM teddy (77)	VGGT	17.977	0.345	0.664	6.571	7121	4286
nuScenes (33)	DA3	30.612	31.826	3.993†	7.023	11569	3639
TartanAir (320)	DA3	29.824	0.452	0.657†	8.202	9970	4286
TUM observed (155)	DA3	31.134	0.447	0.639†	8.083	9970	4286
TUM far (15)	DA3	23.411	0.474	0.746†	7.200	9970	4286
TUM teddy (77)	DA3	20.671	0.406	0.664†	6.188	9970	4286

I PAIRED KNOWN-K MECHANISM STUDY

Complete-roster paired replay. We evaluate all 18 nuScenes DEV windows from six logs and all 256 TartanAir DEV windows from one environment with both MapAnything and Pi3X. These 548 backbone–window replays revisit the 274 DEV inputs for mechanism analysis. Each replay retains six outputs, yielding 3288 scored reconstructions with zero native fallbacks. All six arms share the same frozen native depth and cameras, fixed known intrinsics, RGB correspondences, anchors, and scoring support. Predictions for a complete cohort are saved before its target depths or poses are opened. Means retain equal physical-group weights, as in the primary study.

The MapAnything/TartanAir cell reuses archived native predictions. The other three cells regenerate native predictions at the stored evaluation grid, and all six controls are reevaluated together with common RGB-match caches. Table 19 is therefore a self-contained paired study. MapAnything receives RGB and known intrinsics; Pi3X receives RGB only. Both refiners use the same known intrinsics. Released source interfaces and strict checkpoint loading are retained, with FP32/TF32 inference for MapAnything and BF16 autocast with TF32 disabled for Pi3X. The scalar-evidence supplement records the replay protocol and runtime implementation.

Strict removal of temporary fitting freedom. The no-nuisance arm holds $\ell = 0$ throughout Stage 1 and excludes the nuisance variables from Adam. Matching, weights, camera and persistent-depth parameterizations, priors, and the 320-step budget are unchanged. Its own fitted depth coefficients and cameras initialize an 80-step depth-only refit, using fresh Adam state and the original native depth as the residual basis. Thus both full NativeRefine and no-nuisance+refit use 320+80 updates. The corresponding Stage 1 outputs are also retained.

Direct emission of fitted nuisance. This control starts from the full method's Stage 1 state. Within each view, fitted log-depth nuisance values are interpolated piecewise-linearly over their source-ray coordinates, with nearest-neighbor extension outside the convex hull or for degenerate support. Views without source rays receive zero log correction. The interpolated field multiplicatively corrects the Stage 1 persistent depth; the fitted cameras are retained exactly. Joint and direct-emission outputs use 320 updates. NativeRefine instead emits its bounded, mean-preserving persistent field after the additional 80-step fixed-camera refit.

What each representation contributes. NativeRefine improves both primary reconstruction errors over native predictions in all four cells, and fixed-camera refitting improves point error over centered-joint output in all four. Temporary fitting freedom delivers the strongest camera gains on TartanAir; constrained emission gives the strongest point-error gains over direct emission on nuScenes. Output-policy rankings vary across cells: direct emission leads both primary errors for MapAnything/TartanAir, while Pi3X/TartanAir favors no-nuisance fitting on those endpoints and the full method on rotation and SA translation. The shared Stage 1 cameras of full, joint, and direct-emission outputs make their shared-scale translation differences attributable to the depth-derived scoring scale. Together these controls separate camera fitting, dense correction, and the exact native-statistic preservation of the emitted representation.

All 548 replays complete on the same 16 GB RTX 5070 Ti. Observed full-refiner means range from 6.48 to 6.93 s per window across the four cells; these timings cover the 320+80 optimization updates and exclude native inference, matching, and scoring.

Table 19: Known-K controls: equal-group means per block. P: shared-scale relative point error; T: shared-scale translation; R: rotation; SA-T: independently scale-aligned translation. Bold marks blockwise column minima, including rounded ties. Direct emission and joint-only controls use 320 steps; NativeRefine and no-nuisance + refit use $320 + 80$. There are 548 paired backbone–window replays on 274 DEV inputs.

Method	$P \downarrow$	Shared T (m)↓	R (deg)↓	SA-T (m)↓	AbsRel↓
MapAnything / nuScenes (DEV, $n = 18$)					
Native	0.0903	0.5309	1.4043	0.4013	0.0836
Centered joint	0.0902	**0.4392**	**0.9301**	0.2904	0.0835
No-nuisance joint	0.0874	0.4449	0.9416	**0.2783**	0.0809
No-nuisance + refit	0.0874	0.4449	0.9416	**0.2783**	0.0809
Direct emission	0.0972	0.5465	**0.9301**	0.2904	0.0902
NativeRefine	**0.0872**	0.4510	**0.9301**	0.2904	**0.0808**
MapAnything / TartanAir (DEV, $n = 256$)					
Native	0.2761	0.0559	0.4563	0.0377	0.2227
Centered joint	0.2746	0.0198	**0.1374**	**0.0065**	0.2227
No-nuisance joint	0.2732	0.0336	0.2514	0.0167	0.2215
No-nuisance + refit	0.2732	0.0336	0.2514	0.0167	0.2215
Direct emission	**0.2487**	**0.0165**	**0.1374**	**0.0065**	**0.2020**
NativeRefine	0.2721	0.0192	**0.1374**	**0.0065**	0.2208
Pi3X / nuScenes (DEV, $n = 18$)					
Native	0.0623	0.3594	1.1052	0.3016	0.0579
Centered joint	0.0623	**0.2977**	0.7815	0.2184	0.0579
No-nuisance joint	0.0611	0.3066	**0.7115**	**0.2086**	0.0568
No-nuisance + refit	0.0611	0.3066	**0.7115**	**0.2086**	0.0568
Direct emission	0.0756	0.4853	0.7815	0.2184	0.0701
NativeRefine	**0.0607**	0.3097	0.7815	0.2184	**0.0564**
Pi3X / TartanAir (DEV, $n = 256$)					
Native	0.1750	0.0255	0.1567	0.0148	0.1413
Centered joint	0.1741	0.0146	**0.0642**	**0.0032**	0.1413
No-nuisance joint	**0.1735**	0.0106	0.0892	0.0064	**0.1410**
No-nuisance + refit	**0.1735**	0.0106	0.0892	0.0064	**0.1410**
Direct emission	0.1944	**0.0076**	**0.0642**	**0.0032**	0.1580
NativeRefine	0.1737	0.0140	**0.0642**	**0.0032**	**0.1410**