GAUSSROPE: RETHINKING 3D ROTARY ENCODING WHEN GEOMETRY IS UNCERTAIN

Anonymous authors
Paper under double-blind review

ABSTRACT

3D rotary position embeddings (RoPE) use camera poses and scene depths that can be uncertain. Position errors can distort the relative phases on which attention relies, while geometry refined inside the network cannot inform subsequent encodings. We introduce GaussRoPE, which connects rotary encoding, uncertainty propagation, and layerwise refinement through a unified structured Gaussian geometry state. Propagation maps pose and depth uncertainty into rotary phases, while cross-token shared camera errors are implemented through token-wise rotations. This shared-error sampling preserves the modeled phase correlations and remains compatible with standard attention, without requiring explicit pairwise uncertainty tensors. Visual features update the geometry and its uncertainty, enabling refined states to inform subsequent encodings. We evaluate GaussRoPE across five host models spanning novel-view synthesis, Gaussian reconstruction, 3D object detection, and depth estimation. Relative to the strongest competing encodings in our evaluation, the full GaussRoPE framework improves PETR mAP by approximately 28% under both original and perturbed inputs. Under geometric perturbations, it also improves MVSplat PSNR by 0.48 dB and reduces UniMatch AbsRel by 7.4% relative to competing methods under the same equally weighted pose/depth/joint protocol.

1 INTRODUCTION

A robot searching a room may see only part of a chair from its current position, while an earlier view shows the rest. Recognizing and locating the chair then involves relating the two observations in 3D through the robot's estimated poses and depths. The same need to combine complementary views arises well beyond robotics, in multi-view 3D perception (Liu et al., 2022), novel-view synthesis (Jin et al., 2025), and feed-forward Gaussian reconstruction (Chen et al., 2024). Many recent models for these tasks are built on attention, which by itself does not know where each token lies in 3D. This raises the question of how camera and scene geometry should guide interactions between tokens from different views.

Rotary position embeddings (RoPE) offer an appealing structure for these interactions. RoPE rotates queries and keys separately so that the positional factor in their dot product depends on relative displacement, with multiple frequencies capturing spatial variation at different scales (Su et al., 2024). Because the rotations act on individual tokens, RoPE preserves the standard attention interface. Building on these properties, recent work extends RoPE to 2D image coordinates (Heo et al., 2024), multidimensional coordinates (Schenck et al., 2025), ray-based depth hypotheses (Xie et al., 2026), and uncertain predicted depth (Wu et al., 2026), while camera-based encodings apply relative camera transforms in a similar token-side form (Miyato et al., 2024; Li et al., 2025). In all these methods, however, the resulting interactions depend on the geometry used to construct the encodings.

An image observation can be viewed as a camera ray and an endpoint located by depth. Both are uncertain: consumer depth errors vary with acquisition conditions and surface properties (Khoshelham & Oude Elberink, 2012), while camera rotations and translations retain calibration or tracking errors (Yeshwanth et al., 2023). Network-predicted geometry is also uncertain under ambiguous or incomplete observations (Wang et al., 2024a; Ye et al., 2025). Pose errors perturb a view's rays together; individual depth errors displace endpoints along those rays. These shared and local errors

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.

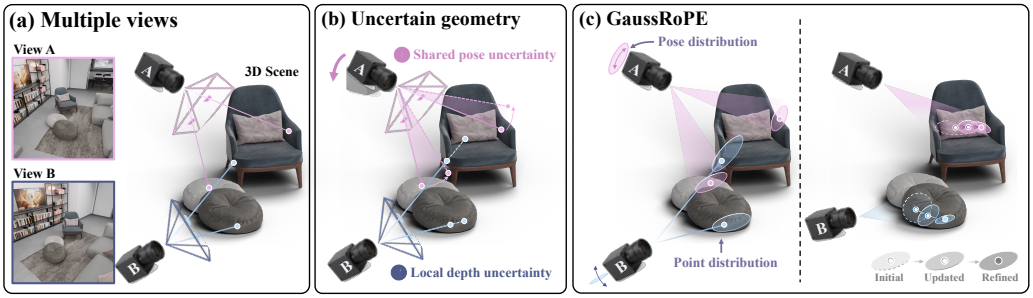


Figure 1: Overview of uncertain geometry for 3D RoPE. (a) Multiple views observe a shared 3D scene. (b) Shared pose errors move a camera's rays together, while local depth errors move points along their rays. (c) GaussRoPE captures both uncertainties through rotary encoding and layerwise geometry updates.

pass through the encoding into rotary phases and can distort attention between views. Visual features provide evidence for refining the same geometry and its uncertainty.

Handling such geometry in 3D RoPE raises three challenges. First, these errors perturb rays and points in different ways. Their effect on each rotary phase also depends on coordinate transforms and encoding frequencies, with higher frequencies responding more strongly to the same error, so a single error magnitude does not describe how each phase is perturbed. Second, a camera error affects all tokens in a view together and thus correlates their phases. Per-token error scales do not capture this correlation, yet storing it for every token pair adds cost quadratic in the number of tokens; the challenge is to preserve it while keeping the standard attention interface. Third, when later layers refine the geometry, subsequent rotary encodings should use the uncertainty of the refined geometry. Since the refined geometry combines the input with network predictions of different reliability, its uncertainty needs to be updated along with it.

We introduce *GaussRoPE*, which addresses these challenges with a single structured Gaussian state over camera poses and depths. Ray–point descriptors are constructed from this observation model, and its shared pose and local depth errors propagate into their rotary phases through coordinate transforms and encoding frequencies. Shared-error sampling then applies this uncertainty through token-side rotations, which preserve the modeled correlations across tokens without explicit pairwise uncertainty tensors and keep the standard attention interface. Finally, layerwise refinement combines geometry predicted from visual features with the source geometry according to their estimated uncertainty, updating both the geometry and its uncertainty in this state for subsequent encodings (Figure 1). We evaluate GaussRoPE on five host models across multi-view tasks and further analyze the encoding design and the behavior of the refined geometry. Our contributions are:

- **A unified framework for 3D rotary encoding.** A structured Gaussian formulation connecting ray–point RoPE, propagation of shared and local geometry errors, and layerwise geometry updates.
- **Correlated uncertainty within standard attention.** Shared-error sampling with token-side rotations, which preserves the phase correlations induced by shared camera errors within each phase channel, without explicit pairwise uncertainty tensors.
- **Layerwise geometry refinement.** An update that combines geometry predicted from visual features with the source geometry, refining both the geometry and its uncertainty for subsequent rotary encodings.

2 Related Work

Geometry-aware multi-view attention. Geometry conditions multi-view transformers through ray tokens in LVSM, 3D feature coordinates in PETR, and object-centric temporal information in StreamPETR (Jin et al., 2025; Liu et al., 2022; Wang et al., 2023). Within attention, GTA factorizes relative feature transformations into token-side operations, including value aggregation; PRoPE extends this construction to camera intrinsics and extrinsics while retaining fused-kernel compat-

ibility (Miyato et al., 2024; Li et al., 2025). Alongside these general transformations, RoPE uses rotations to encode relative positions (Su et al., 2024). RoPE-ViT explores mixed spatial frequencies, STRING develops multidimensional rotary encodings, and RoRE encodes camera rays across perspective, fisheye, and RGB–thermal settings (Heo et al., 2024; Schenck et al., 2025; Griffiths & Dansereau, 2026). URoPE distributes depth anchors along rays across heads and can incorporate provided depth; it also evaluates noisy camera parameters (Xie et al., 2026). RayRoPE predicts ray depths and encodes camera centers, projected coordinates, and inverse depths in the query camera frame (Wu et al., 2026). Our focus is uncertain pose and depth in relative rotary phases, rather than geometric conditioning or factorization alone. Appendix A summarizes the encoding representations.

Probabilistic geometry and uncertainty-aware encoding. Probabilistic SLAM models poses and landmarks jointly; EKF-SLAM propagates their means, covariances, and cross-correlations (Durrant-Whyte & Bailey, 2006). These covariances describe estimation errors, unlike the rendering extent represented by 3D Gaussian Splatting (Kerbl et al., 2023). Gaussian maps can also carry geometric, semantic, and appearance uncertainty for navigation (Gao et al., 2026). For encoding distributions, mip-NeRF integrates sinusoidal features over Gaussian pixel footprints (Barron et al., 2021). RayRoPE already computes expected rotary features over uncertain projected positions; its relative-encoding derivation assumes independent positions and leaves camera-matrix uncertainty for future work (Wu et al., 2026). Our focus is propagating shared camera and local depth errors into relative rotary phases without explicit pairwise uncertainty tensors. The structured model retains specified dependencies and the standard attention interface, but does not represent a full joint scene posterior.

Geometry estimation and refinement. Geometry may come from tracking or learned estimation; SLAM benchmarks explicitly measure trajectory drift and pose error (Sturm et al., 2012). MVSplat estimates depth from posed views using cost volumes, while NoPoSplat predicts Gaussians from unposed images and supplied intrinsics (Chen et al., 2024; Ye et al., 2025). DUSt3R uses pointmap confidence for regression and alignment; CUT3R updates persistent state and reads out geometry, confidence, and poses (Wang et al., 2024a; 2025a). RayRoPE already predicts depth and uncertainty at successive attention layers (Wu et al., 2026). We connect pose-and-depth updates to the structured error state used by subsequent rotary encodings. These internal revisions serve the host task; their utility and agreement with reference geometry are distinct evaluation questions.

3 METHOD

3.1 OVERVIEW

GaussRoPE represents image observations as camera rays and depth-defined endpoints. Camera pose errors move a view's rays together; local depth errors move individual endpoints along those rays. One Gaussian state G_ℓ represents this geometry and its shared and local errors. Ray–point RoPE encodes descriptors of this state, and uncertainty propagation carries the same errors into their rotary phases. Attention fuses visual evidence; readouts and source–proposal fusion update the geometry and uncertainty together to form $G_{\ell+1}$ (Figure 2). The index ℓ denotes designated feedback positions. Hosts may fix selected variables; their task heads produce the outputs used to assess utility.

The derivation below concerns camera-associated image tokens, with Pi3X as the concrete prediction instance. Its frozen predictor F_0, supervision and Q/K/V/output transforms are instance choices. Query roles depend on the task: EmbodiedScan uses 3D point queries. The host interfaces are specified in Appendix C and Table 8.

3.2 PROBABILISTIC 3D GEOMETRY

For image token i belonging to camera $c(i)$, let $T_c = (R_c, C_c)$ be a camera-to-world pose with rotation R_c and camera center C_c, r_i a unit camera-frame ray, and $\rho_i > 0$ its radial distance. The corresponding world direction and point are $D_i = R_{c(i)} r_i$ and $X_i = C_{c(i)} + \rho_i D_i$. For a cali-

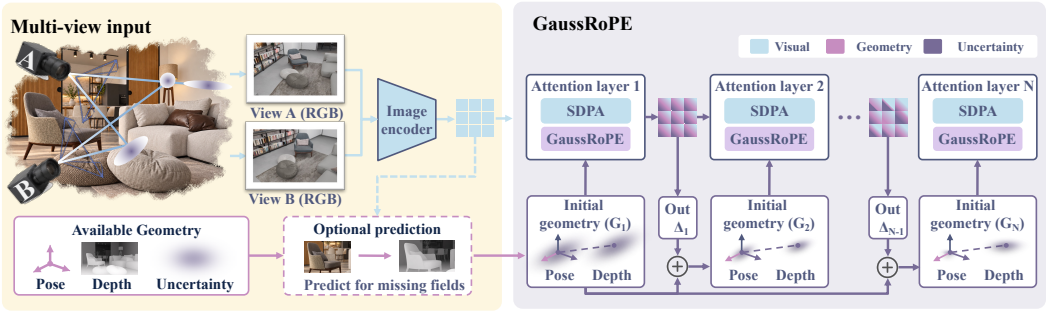


Figure 2: GaussRoPE overview. A Gaussian geometry state supplies each RoPE stage; feature readouts update it for later layers. The fixed source G_0 anchors updates; host heads produce task outputs.

brated pixel u_i, r_i is the normalized $K_c^{-1}u_i$. Patch representatives and camera-Z conversion are in Appendix C.1; we omit stage indices below.

One state, shared and local errors. We place a local Gaussian error model on each camera pose and each token's log radial distance:

$$\delta\xi_c \sim \mathcal{N}(0, P_c), \qquad \delta\eta_i \sim \mathcal{N}(0, \sigma_{\eta,i}^2), \qquad \eta_i = \log \rho_i. \tag{1}$$

The six-dimensional pose error $\delta\xi_c$ jointly perturbs the origins, directions and endpoints of camera c's rays. The local depth error $\delta\eta_i$ moves endpoint X_i along D_i with the pose fixed. Each image token reuses these errors wherever it participates in attention. Distinct cameras and distinct token-depth errors are modeled as independent. Linearization yields a local 3D Gaussian for each point, with cross-token covariance inherited from shared camera errors (Appendix D.1). Thus the points are not independent Gaussians. Unlike the rendering extent in Gaussian Splatting (Kerbl et al., 2023), these covariances describe estimation error; no splatting operation is required.

3.3 Uncertainty Propagation and Rotary Attention

Ray–Point Rotary Encoding. We construct ray and point descriptors from this same observation model. In the image-token realization, camera c supplies a common reference frame: $(C_i^{(c)}, D_i^{(c)}, X_i^{(c)})$ are obtained using $T_c^{-1}T_{c(i)}$. This frame choice does not define the task's query type. Ray and point channel groups occupy distinct rotary pairs within each head and encode

$$g_{i,\text{ray}}^{(c)} = [D_i^{(c)};\ C_i^{(c)} \times D_i^{(c)}/s], \qquad g_{i,\text{point}}^{(c)} = X_i^{(c)}/s, \qquad \phi_{i,h,b}^{(c)} = \omega_{h,b}^{\top} g_i^{(c)}, \tag{2}$$

Here s is a fixed scene scale, h an attention head, b a two-channel rotary pair, and $\omega_{h,b}$ its frequency vector. The ray descriptor couples camera pose and ray direction through an oriented line; the point descriptor uses the 3D endpoint $X_i = C_{c(i)} + \rho_i D_i$. The descriptors share pose variables, with depth determining the endpoint. A common world-frame $\mathrm{SE}(3)$ transform leaves the relative geometry unchanged (Appendix D.1.2).

To propagate uncertainty, index camera and token-depth error sources by o, with $\epsilon_o \sim \mathcal{N}(0, P_o)$. In one frame, head, and rotary pair, the local phase model and relative-phase variance are

$$\phi_i \simeq \bar{\phi}_i + \sum_o a_{i,o}^{\top}\epsilon_o, \qquad v_{ij} = \sum_o (a_{j,o} - a_{i,o})^{\top} P_o (a_{j,o} - a_{i,o}). \tag{3}$$

The sensitivity $a_{i,o}$ includes the reference-frame transformation. Relative uncertainty depends on covariance as well as individual phase variances. A shared error cancels only when its sensitivities agree: consistent same-camera perturbations cancel in $T_c^{-1}T_c$, whereas independent depth errors remain. Figure 7 in the appendix illustrates the translation-only case.

From phase to attention. Fourier expectations encode distributions, including Gaussian regions in mip-NeRF (Barron et al., 2021) and uniform projected-position components in RayRoPE (Wu et al., 2026). Our linear Gaussian model gives $\mathbb{E}[\exp(\mathrm{i}(\phi_j - \phi_i))] = \exp(\mathrm{i}(\bar{\phi}_j - \bar{\phi}_i) - v_{ij}/2)$. This

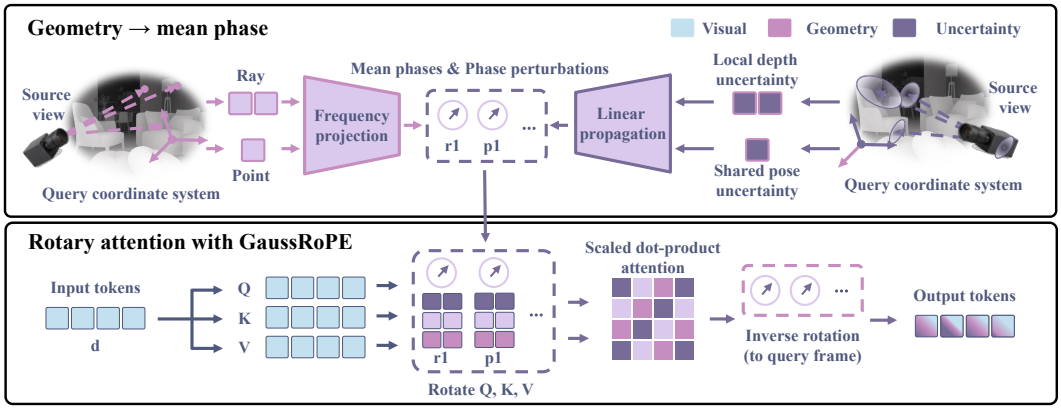


Figure 3: Geometry-to-attention mapping. Shared pose and local depth errors perturb ray/point phases before token-side rotations. Q/K/V rotation and output restoration depict the Pi3X instance; other hosts may retain native V/O.

identity describes a phase factor, not an expected softmax output. Figure 3 connects geometry, phase variation, and the sampled operator below.

A sampled attention operator. Independently sampled token phases would lose shared-error covariance, while explicit pairwise factors add storage and preparation cost. To retain the specified dependencies with token-side operations, we sample the errors in Equation 3, reusing camera samples across their tokens and depth samples across query/key roles. Sharing includes source/reference-camera errors within each descriptor group, head, and rotary pair. These channels use independent streams, preserving their modeled cross-token dependence rather than one jointly sampled 3D scene. For the resulting block-diagonal transform $\mathcal{R}_i$ in the reference frame and head width d_h, the Pi3X attention form is

$$\alpha_{ij} = \mathrm{softmax}_j\left(\frac{(\mathcal{R}_i q_i)^\top (\mathcal{R}_j k_j)}{\sqrt{d_h}}\right), \qquad y_i = \mathcal{R}_i^{-1} \sum_j \alpha_{ij} \mathcal{R}_j v_j. \tag{4}$$

The projections q_i, k_j, v_j come from visual features, and y_i returns to the feature stream. Inspired by RayRoPE (Wu et al., 2026), we rotate values and restore the output, making each message $\mathcal{R}_i^{-1}\mathcal{R}_j v_j$ depend on relative geometry. LVSM, MVSplat and UniMatch retain native V/O and rotate only Q/K. The Pi3X path uses one sampled phase bank and one attention evaluation per stage, including inference (Appendix C.3).

Shared uncertainty without pairwise expansion. We do not materialize v_{ij}: within each rotary pair, $(R(\phi_i)q_i)^\top (R(\phi_j)k_j) = q_i^\top R(\phi_j - \phi_i)k_j$. Reused samples and ordinary dot products form relative phases implicitly. Token-side geometric transforms already appear in GTA and PRoPE (Miyato et al., 2024; Li et al., 2025). Here they retain the specified sampled dependencies. In the Pi3X realization, grouping image queries by camera yields standard operands without a pairwise uncertainty bias, compatible with FlashAttention (Dao et al., 2022). For batch size B, C cameras, N tokens and H heads, phase storage is $O(BCNHd_h)$ versus $O(BHN^2d_h)$ for dense pairwise factors. Token-side storage is linear in N for fixed C; query-view replication and quadratic attention arithmetic remain. Section 4.4 measures factor preparation in a separate-head reference, not the runtime of this within-head layout.

3.4 Layerwise Updates of RoPE Geometry

Each feedback stage reads fused features to propose geometry and estimate separate source/proposal error scales. The source means stay anchored at G_0, while their predicted scales may change. Both the fused means and uncertainty are used in the next propagation step; updating means alone would leave reliability tied to an earlier state.

Fuse without assuming independent evidence. Because source and proposal may reuse images, we use fixed-half covariance intersection (Julier & Uhlmann, 1997) in a common chart. For means m_s, m_v and positive diagonal covariances P_s, P_v,

$$P_f^{-1} = \tfrac{1}{2}P_s^{-1} + \tfrac{1}{2}P_v^{-1}, \qquad m_f = P_f\left(\tfrac{1}{2}P_s^{-1}m_s + \tfrac{1}{2}P_v^{-1}m_v\right). \tag{5}$$

The visual mean gain is $k = P_s/(P_s + P_v)$ coordinatewise. Equal precision coefficients make the rule symmetric without another learned weight; they do not fix k to one half. The same scales determine the correction and fused uncertainty. The rule uses learned error scales to weight the two estimates; their training interpretation is specified below.

Pi3X realization. Pi3X extends π^3 (Wang et al., 2025b). Valid observations initialize G_0, with missing fields supplied by a frozen native estimate F_0. Patch and camera-pooled features provide depth and pose proposals. Fusion uses log-depth and a source-anchored local $\mathrm{SE}(3)$ chart; pose covariance is transported to the updated tangent. Descriptors and phase sensitivities are recomputed at $G_{\ell+1}$ before the next RoPE stage. Appendix C specifies anchors, readouts and transport.

Closing the structured state. We retain Equation 1's structure after each update, omitting additional cross-camera, cross-token depth, and pose–depth correlations introduced by visual fusion. Covariance intersection handles source–proposal dependence within each variable, yielding a structured state approximation for subsequent encoding.

3.5 Training and Inference

Objectives and gradient paths are host-specific: PETR uses task losses, whereas LVSM freezes geometry-module parameters during transfer while retaining forward state updates (Appendix B.1). Pi3X supervises depth, rotation, and translation, with Gaussian NLL losses for source/proposal error scales. Detached residuals and features make NLL fit scales to training-reference residuals without moving proposal means or backbone features; task gradients still pass through fusion and attention. Fused uncertainty follows Equation 5 without a separate NLL target. Appendix C.3 gives the losses and schedule. Inference uses images, available geometry and observation masks, without reference geometry or corruption labels.

4 Experiments

Our evaluation separates complete-framework performance, a joint ablation of its coupled propagation–update loop, and controlled studies of specific design choices. We also measure uncertainty-propagation costs.

4.1 Experimental Setup

Table 1 covers direct view synthesis with LVSM and Gaussian reconstruction with MVSplat on RealEstate10K (RE10K) (Jin et al., 2025; Chen et al., 2024; Zhou et al., 2018), PETR on nuScenes (Liu et al., 2022; Caesar et al., 2020), EmbodiedScan on ScanNet (Wang et al., 2024b; Dai et al., 2017), and UniMatch on SUN3D (Xu et al., 2023; Xiao et al., 2013). Baselines are the native host, PRoPE, URoPE, RoRE, RayRoPE and GTA (Li et al., 2025; Xie et al., 2026; Griffiths & Dansereau, 2026; Wu et al., 2026; Miyato et al., 2024). GaussRoPE-static uses our ray–point encoding of camera rays and depth-defined endpoints; Full adds the coupled uncertainty-propagation and geometry-update loop. Their comparison is a joint system ablation. Main results compare single-seed, fixed-budget host adaptations; controlled studies examine specific choices. Training, input access and attention interfaces are specified in Appendix B.

4.2 Task Performance under Uncertain Geometry

Original uses supplied geometry without added perturbations. Noisy preserves each adaptation's perturbable encoding inputs (Table 6). RayRoPE, Static and Full share pose/depth/joint conditions and weights. P adapters lack a perturbed encoding-depth input; adding a depth pathway would change the baseline. Noisy summarizes each native interface, while Table 10(b) compares methods on common pose-only conditions.

Table 1: **Downstream performance with uncertain geometry.** Each host reports one primary task metric under original inputs (Orig.) and corrupted geometry (Noisy). Noisy protocols differ by host and method (Appendix B.1, Table 6); higher is better except for AbsRel.

	View synthesis		Reconstruction		3D detection				Depth	
	LVSM RE10K PSNR↑		MVSplat RE10K PSNR↑		PETR nuScenes mAP↑		EmbodiedScan ScanNet60 AP_{25}↑		UniMatch SUN3D AbsRel↓	
Method	Orig.	**Noisy**	Orig.	**Noisy**	Orig.	**Noisy**	Orig.	**Noisy**	Orig.	**Noisy**
Native	17.24	17.22	18.29	17.62	1.24	1.23	11.71	11.51	1.025	2.520
PRoPE	22.46	22.43	18.28	17.74	0.51	0.51	12.39	11.96	1.201	2.672
URoPE	**24.36**	**23.56**	18.13	17.70	1.02	1.02	10.74	10.63	0.952	2.108
RoRE	22.68	22.21	18.16	17.80	3.70	3.65	11.15	11.13	0.985	2.208
RayRoPE	22.11	21.72	18.21	17.91	4.05	4.02	12.01	11.90	**0.805**	1.511
GTA	22.49	22.46	17.66	17.26	0.48	0.48	12.24	11.67	1.245	2.442
GaussRoPE-static	23.48	22.06	18.43	18.11	5.15	5.12	11.46	11.38	1.008	1.815
GaussRoPE	23.45	22.89	**18.59**	**18.39**	**5.18**	**5.15**	**12.81**	**12.43**	0.963	**1.399**

Static uses RoPE only; GaussRoPE includes uncertainty propagation and geometric updates. mAP and AP_{25} are percentages; AP_{25} uses test-time augmentation (TTA). Noisy preserves each adaptation's perturbable inputs; RayRoPE, Static and Full share the same pose/depth/joint sets and weights. Bold/underline: best/second-best displayed values, not a common-condition ranking across different protocols; –: incomplete or unavailable. Single-seed host adaptations; protocols: Appendix B.1. Common pose-only metrics: Table 10(b).



Figure 4: Selected multi-task outputs under pose perturbations (P3D0). Rows share inputs, views and crops; Static and Full are shown separately. Numbers are crop PSNR, object 9DoF IoU, or valid-pixel AbsRel, not benchmark scores. Solid/dashed boxes denote prediction/GT; gray depth pixels are invalid. Display protocol: Appendix D.3.3.

Static already achieves most of PETR's gain over RayRoPE; Full adds 0.03 percentage points in both conditions. Full improves all five primary Noisy metrics and four Original metrics over Static; LVSM PSNR changes from 23.48 to 23.45 dB. URoPE leads LVSM PSNR; RayRoPE leads Original UniMatch AbsRel. Full improves all reported pose-only metrics over Static; other rankings remain task- and metric-dependent (Table 10(b); Appendix D.3). These are complete host adaptations, including training and geometry interfaces (Figure 4).

Response to perturbation strength. Figure 5 separates MVSplat's pose, depth and joint responses. Full leads each displayed condition; its joint PSNR advantage over Static is 0.24–0.42 dB. Pose noise has a larger effect than the tested local encoding-depth noise, showing that robustness varies with the error source (complete grid and perturbations: Appendix D.3.2).

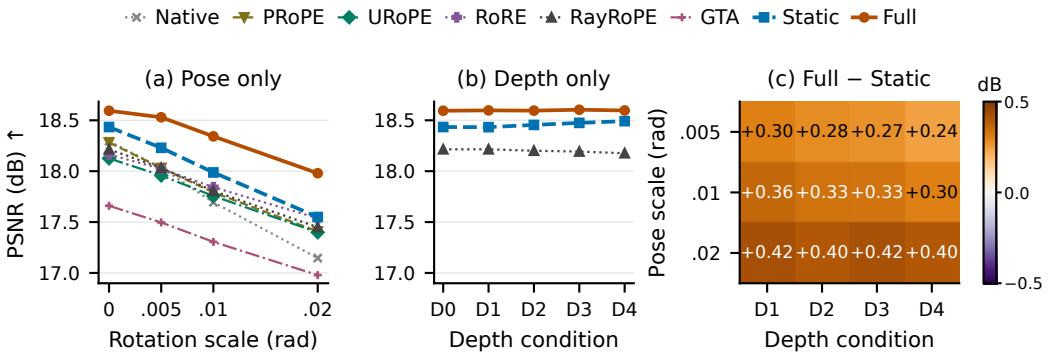


Figure 5: MVSplat under pose, depth and joint perturbations (38 scenes, one seed). Pose curves include all eight methods; depth curves include the three applicable methods. The grid shows Full minus Static. Conditions and absolute scores: Appendix D.3.2.

4.3 Controlled Studies of Encoding and Feedback

Controlled studies examine shared-error structure, phase propagation, and feedback within the loop. **Why model shared camera errors?** Table 2 tests two MVSplat variants with matched initialization, inputs and training budgets. The control makes source-pose errors independent across tokens and from query-coordinate errors, removing common-mode cancellation. Retaining camera sharing improves perturbed PSNR by 0.319 dB (95% paired-scene interval: [0.058, 0.596]), with higher SSIM and lower LPIPS. This supports shared-error modeling in 3D RoPE. Both variants learn their scales; Perturbed-19 averages conditions, unlike Table 1's type-balanced Noisy (Appendix E).

Table 2: **Shared-camera error ablation in MVSplat.** Matched training; 38 scenes, one seed. Perturbed-19 averages all 19 noisy conditions (Appendix E).

	Original			Perturbed-19		
Pose error structure	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
Sharing removed	18.144	0.603	0.380	17.765	0.580	0.394
Camera-shared	**18.402**	**0.606**	**0.345**	**18.084**	**0.590**	**0.356**

From geometry uncertainty to rotary phases. Pi3X's relative-phase factor error against nonlinear Monte Carlo rises from 0.00293 at nominal noise to 0.01875 at four times that scale (Appendix C.5). Distinct from sharing, sampling improves PSNR/SSIM but increases LPIPS in three-seed NVS controls; Pi3X main paired sampled/mean-phase comparison has a below-zero rotation interval, while depth and center intervals include zero (Appendices D.3.4 and B.3).

Task outputs from geometry updates. Table 3(a) compares separately trained prediction/fusion schemes, not an isolated uncertainty ablation: Full lowers all three displayed joint-perturbation errors relative to Direct update (Appendix B.3). In the separately recorded evaluation in (b), updated geometry yields lower depth and pose error point estimates at fixed Full weights. All blocks and readouts execute; only the consumed geometry changes.

Table 3: **Pi3X prediction and feedback controls.** Joint perturbations on 24 development spaces; one training seed. Panel (b) reports a separate fixed-checkpoint evaluation. Protocol: Appendix B.3.

(a) Trained schemes (500 updates)

Variant	AbsRel ↓	Rotation (°) ↓	Center (mm) ↓
Direct update	0.0760	1.894	54.24
Full	**0.0403**	**0.795**	**28.13**

(b) Fixed Full weights (0 updates)

Feedback input	AbsRel ↓	Rotation (°) ↓	Center (mm) ↓
Normal feedback	**0.0403**	**0.845**	**30.66**
Always read G_0	0.0422	0.953	34.64

4.4 Computational Cost

We measure the time and memory saved by representing relative-phase uncertainty through token-side rotations. Figure 6 fixes the geometry and shared error samples, comparing explicit pairwise factors with their token-side factorization. Holding the phases fixed isolates the cost of their representation. For four views with $P = 512$ tokens each, preparation drops from 27.71 to 2.31 ms and from 18,444 to 63.02 MiB (A100 40 GB, FP32; 30 runs). Token-side rotations also retain the standard Q/K/V interface, allowing FlashAttention to handle both forward and backward computation. Appendix C.4 gives the separate-head benchmark protocol and complements it with actual fused-kernel execution and full-training measurements.

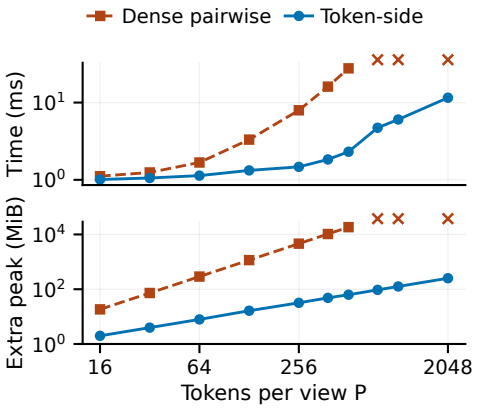


Figure 6: Time and extra memory for factor preparation. Crosses mark OOM.

5 Limitations

GaussRoPE uses a structured, locally unimodal Gaussian approximation to geometry uncertainty. This representation captures camera-shared pose errors and token-local depth errors, but does not model arbitrary multimodal scene hypotheses or a full joint geometry posterior. Its first-order propagation is most suitable for moderate uncertainties and may become less accurate under large rotations, broad depth ambiguity, or strongly nonlinear camera models. The current formulation also assumes fixed intrinsics.

The predicted uncertainty scales are learned from training residuals and should therefore be interpreted as task- and distribution-dependent reliability estimates rather than universally calibrated physical noise levels. Experiments use several host-specific integrations; accordingly, the main results evaluate end-to-end utility under the corresponding protocols, while broader component attribution and cross-domain validation remain important directions for future work. Finally, token-side factorization reduces uncertainty-representation overhead but does not remove the quadratic attention cost, and the additional geometry updates introduce moderate computation.

6 Conclusion

We presented GaussRoPE, a structured framework for geometry-aware rotary encoding under uncertain multi-view geometry. It models camera-shared pose and token-local depth uncertainty, propagates them to ray–point phases, and refines the state layerwise from visual features. Across five host integrations, Full outperforms RoPE-only Static on all five primary noisy endpoints under their respective protocols. These results indicate that uncertainty-aware phase propagation and geometry feedback can improve robustness to perturbed geometry. Controlled studies provide evidence for shared-camera error modeling and layerwise feedback, while token-side factorization reduces the measured cost of relative-phase preparation without changing the standard attention interface.

AI USE STATEMENT

We used generative AI tools to assist with manuscript writing and language polishing, retrieve relevant literature and discover related work. The research assistance included code development, and program execution. The authors take responsibility for the final content of the paper, including its mathematical claims, implementation, experimental results, and conclusions.

REFERENCES

Jonathan T. Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P. Srinivasan. Mip-NeRF: A multiscale representation for anti-aliasing neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5855–5864, 2021.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11618–11628, 2020. doi: 10.1109/CVPR42600.2020.01164.

Yuedong Chen, Haofei Xu, Chuanxia Zheng, Bohan Zhuang, Marc Pollefeys, Andreas Geiger, Tat-Jen Cham, and Jianfei Cai. MVSplat: Efficient 3D gaussian splatting from sparse multi-view images. In *Computer Vision – ECCV 2024*, pp. 370–386, 2024. doi: 10.1007/978-3-031-72664-4_21.

Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2432–2443, 2017. doi: 10.1109/CVPR.2017.261.

Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. FlashAttention: Fast and memory-efficient exact attention with IO-awareness. In *Advances in Neural Information Processing Systems*, volume 35, pp. 16344–16359, 2022. doi: 10.52202/068431-1189.

Hugh Durrant-Whyte and Tim Bailey. Simultaneous localization and mapping: Part I. *IEEE Robotics & Automation Magazine*, 13(2):99–110, 2006. doi: 10.1109/MRA.2006.1638022.

Jianzhe Gao, Rui Liu, Yuxuan Xu, Tongtong Cao, Yingxue Zhang, Zhanguang Zhang, Sida Peng, Yi Yang, and Wenguan Wang. Uncertainty-aware gaussian map for vision-language navigation. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=LPv59noPAy.

Ryan Griffiths and Donald G. Dansereau. RoRE: Rotary ray embedding for generalised multi-modal scene understanding. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=BR2ItBcqOo.

Byeongho Heo, Song Park, Dongyoon Han, and Sangdoo Yun. Rotary position embedding for vision transformer. In *Computer Vision – ECCV 2024*, pp. 289–305, 2024. doi: 10.1007/978-3-031-72684-2_17. URL https://www.ecva.net/papers/eccv_2024/papers_ECCV/html/1584_ECCV_2024_paper.php.

Haian Jin, Hanwen Jiang, Hao Tan, Kai Zhang, Sai Bi, Tianyuan Zhang, Fujun Luan, Noah Snavely, and Zexiang Xu. LVSM: A large view synthesis model with minimal 3D inductive bias. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=QQBPWtvtcn.

S. J. Julier and J. K. Uhlmann. A non-divergent estimation algorithm in the presence of unknown correlations. In *Proceedings of the 1997 American Control Conference*, volume 4, pp. 2369–2373, 1997. doi: 10.1109/ACC.1997.609105.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3D gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4), 2023. doi: 10.1145/3592433.

Kourosh Khoshelham and Sander Oude Elberink. Accuracy and resolution of Kinect depth data for indoor mapping applications. *Sensors*, 12(2):1437–1454, 2012. doi: 10.3390/s120201437. URL https://pmc.ncbi.nlm.nih.gov/articles/PMC3304120/.

Ruilong Li, Brent Yi, Junchen Liu, Hang Gao, Yi Ma, and Angjoo Kanazawa. Cameras as relative positional encoding. *arXiv preprint arXiv:2507.10496*, 2025.

Yingfei Liu, Tiancai Wang, Xiangyu Zhang, and Jian Sun. PETR: Position embedding transformation for multi-view 3D object detection. In *Computer Vision – ECCV 2022*, pp. 531–548, 2022. doi: 10.1007/978-3-031-19812-0_31.

Takeru Miyato, Bernhard Jaeger, Max Welling, and Andreas Geiger. GTA: A geometry-aware attention mechanism for multi-view transformers. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=uJVHygNeSZ.

Connor Schenck, Isaac Reid, Mithun George Jacob, Alex Bewley, Joshua Ainslie, David Rendleman, Deepali Jain, Mohit Sharma, Kumar Avinava Dubey, Ayzaan Wahid, Sumeet Singh, René Wagner, Tianli Ding, Chuyuan Fu, Arunkumar Byravan, Jake Varley, Alexey A. Gritsenko, Matthias Minderer, Dmitry Kalashnikov, Jonathan Tompson, Vikas Sindhwani, and Krzysztof Marcin Choromanski. Learning the RoPEs: Better 2D and 3D position encodings with STRING. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 53295–53315. PMLR, 2025. URL https://proceedings.mlr.press/v267/schenck25a.html.

Jürgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers. A benchmark for the evaluation of RGB-D SLAM systems. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 573–580, 2012. doi: 10.1109/IROS.2012.6385773. URL https://vision.in.tum.de/_media/spezial/bib/sturm12iros.pdf.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. RoFormer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024. doi: 10.1016/j.neucom.2023.127063.

Benjamin Ummenhofer, Huizhong Zhou, Jonas Uhrig, Nikolaus Mayer, Eddy Ilg, Alexey Dosovitskiy, and Thomas Brox. Demon: Depth and motion network for learning monocular stereo. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5622–5631, 2017. doi: 10.1109/CVPR.2017.596.

Qianqian Wang, Yifei Zhang, Aleksander Holynski, Alexei A. Efros, and Angjoo Kanazawa. Continuous 3D perception model with persistent state. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10510–10522, June 2025a.

Shihao Wang, Yingfei Liu, Tiancai Wang, Ying Li, and Xiangyu Zhang. Exploring object-centric temporal modeling for efficient multi-view 3D object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3621–3631, 2023.

Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. DUSt3R: Geometric 3D vision made easy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20697–20709, 2024a.

Tai Wang, Xiaohan Mao, Chenming Zhu, Runsen Xu, Ruiyuan Lyu, Peisen Li, Xiao Chen, Wenwei Zhang, Kai Chen, Tianfan Xue, Xihui Liu, Cewu Lu, Dahua Lin, and Jiangmiao Pang. EmbodiedScan: A holistic multi-modal 3D perception suite towards embodied AI. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19757–19767, 2024b. doi: 10.1109/CVPR52733.2024.01868.

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. *arXiv preprint arXiv:2507.13347*, 2025b.

Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4): 600–612, 2004. doi: 10.1109/TIP.2003.819861.

Yu Wu, Minsik Jeon, Jen-Hao Rick Chang, Oncel Tuzel, and Shubham Tulsiani. RayRoPE: Projective ray positional encoding for multi-view attention. *arXiv preprint arXiv:2601.15275*, 2026.

Jianxiong Xiao, Andrew Owens, and Antonio Torralba. SUN3D: A database of big spaces reconstructed using SfM and object labels. In *Proceedings of the IEEE International Conference on Computer Vision*, 2013.

Yichen Xie, Depu Meng, Chensheng Peng, Yihan Hu, Quentin Herau, Masayoshi Tomizuka, and Wei Zhan. URoPE: Universal relative position embedding across geometric spaces. *arXiv preprint arXiv:2604.18747*, 2026.

Haofei Xu, Jing Zhang, Jianfei Cai, Hamid Rezatofighi, Fisher Yu, Dacheng Tao, and Andreas Geiger. Unifying flow, stereo and depth estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

Botao Ye, Sifei Liu, Haofei Xu, Xueting Li, Marc Pollefeys, Ming-Hsuan Yang, and Songyou Peng. No pose, no problem: Surprisingly simple 3d gaussian splats from sparse unposed images. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=P4o9akekdf.

Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Nießner, and Angela Dai. ScanNet++: A high-fidelity dataset of 3D indoor scenes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 12–22, 2023. URL https://openaccess.thecvf.com/content/ICCV2023/html/Yeshwanth_ScanNet_A_High-Fidelity_Dataset_of_3D_Indoor_Scenes_ICCV_2023_paper.html.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 586–595, 2018. doi: 10.1109/CVPR.2018.00068.

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

A Extended Related Work

Table 4 complements Section 2 with the geometric objects and depth interfaces of representative attention encodings. It compares representations, not task performance or noise robustness, and does not describe every capability of the host models.

Table 4: Geometry used by representative multi-view attention encodings. The depth column describes the encoding, not all capabilities of its host model. Sources: GTA (Miyato et al., 2024), PRoPE (Li et al., 2025), URoPE (Xie et al., 2026), and RayRoPE (Wu et al., 2026).

Method	Geometry in the encoding	Treatment of scene depth
GTA	Camera extrinsics and image positions	No depth prediction in the encoding.
PRoPE	Relative camera frusta and image positions	No depth prediction in the encoding.
URoPE	Ray points projected into the query view	Fixed anchors across heads; available depths can replace anchors.
RayRoPE	Camera center and ray endpoint in the query frame	Per-layer depth and uncertainty; expected encoding or supplied depth.

B Reproducibility Details

B.1 Task and Input Protocols

The task protocols distinguish available input geometry from references used for supervision or evaluation under each host's protocol. The specification includes the pose and depth sources, active

geometric variables, masks, initialization, and supervision. Synthetic perturbations act on the designated inputs, not on target images or evaluation references. Natural estimation errors and additional perturbations are reported as distinct conditions.

Table 5: Protocols used in the main and supplementary task tables. Each row uses one training seed.

Host	Train / evaluation	Updates; batch	Seed
NVS transfer	RE10K 500 / 100 scenes	2500; 4	0
MVSplat	RE10K 39 / 38 scenes	2500; 2×2	111124
PETR	nuScenes 2396 / 1048 frames	3000; 5	1
StreamPETR	nuScenes 2396 / 1048 frames	7188; 1	0
EmbodiedScan	ScanNet 48 / 12 scenes	720; 8	1
UniMatch	SUN3D 48 validation pairs	1250; 8	111124

NVS uses the DeMoN training corpus (Ummenhofer et al., 2017) for source training (1250 updates), followed by RE10K transfer (2500 updates), with predicted geometry at the target stage. Full uses a smooth bound on the norm of its axis-angle rotation coordinates below π to avoid the chart boundary; its geometry modules are frozen during target training. Source-stage depth access and implementations differ across methods, so this is a method-package comparison. MVSplat keeps the 192-channel, two-head host, additive 2D encoding and no additional camera-ray input embedding; it evaluates 114 targets in 38 development scenes. All MVSplat methods use clean+pose mixed training and pose robustness evaluation. Only RayRoPE and the two GaussRoPE variants additionally use depth-noise training and depth/joint robustness evaluation; the other methods have no perturbed encoding-depth input. UniMatch uses the native 192-channel, two-head depth host and three validation scenes. The Full models for these two hosts use the encoding scale computed from the initial geometry; their geometry modules and auxiliary losses differ from the static models.

EmbodiedScan uses mixed-noise training for *every* main-table method and reports AP_{25} with test-time augmentation (TTA) over the fixed 67 classes. Separate no-TTA diagnostics appear in Table 10; no-TTA does not mean single-view input. Native denotes the attention control with identity geometric transforms. All EmbodiedScan methods use 256 attention channels except RayRoPE, which uses eight heads of 48 channels to satisfy its requirement that each head dimension be divisible by 24. All main-table methods execute three attention passes. Full performs two geometry updates between these passes; it does not add attention passes. GTA uses the same mixed training, seed, and 720-update budget. Its homogeneous GTA adaptation is specified in Table 6(c).

PETR uses mixed-noise training with aligned frequency ranges; RayRoPE retains wider attention. All methods start from the same ImageNet-pretrained ResNet-50 initialization, with seed 1, batch 5 and 3000 updates (15,000 exposures), AdamW learning rate 1.25×10^{-4}, cosine decay and 500 warmup updates. The nuScenes subset has 59 training scenes (2396 frames) and 26 disjoint validation scenes (1048 frames). Sample order, augmentation, noise draws, and shared host initialization were checked across arms. Original and Noisy use the same final mixed-trained checkpoint. Full uses five geometry updates, task losses only, and its specified uncertainty priors; it does not use the initial-geometry scale normalization of MVSplat.

The paired schedule (noise seed 111124) assigns 25% clean, 12.5% pose, 12.5% depth and 50% joint exposures. Depth-inapplicable arms effectively receive 37.5% clean and 62.5% pose exposures. Pose levels use scales $0, 0.005, 0.01, 0.02$ radians; translation scales multiply these by the training-only median camera baseline, 0.677 m. CAM_FRONT stays fixed; one or two other cameras receive scene/camera bias plus frame jitter at 0.1 times the scale. Camera-local errors modify extrinsics and projections; RGB, intrinsics and evaluation references stay fixed. For methods whose encodings use depth, approximately a quarter-region receives log-depth bias, jitter and missingness with (b, σ, r) levels $(0, 0, 0)$, $(0, 0.01, 0)$, $(0.025, 0.01, 0.05)$, $(0.05, 0.025, 0.1)$ and $(0.1, 0.05, 0.2)$; other tokens are preserved. This corrupts predicted encoding geometry, not observed sensor depth. RayRoPE, Static, and Full each evaluate all 4×5 conditions on the same 1048 frames; the other arms evaluate four pose conditions, with depth marked inapplicable. Native pooled mAP/NDS and the evaluated frame sets were checked. GTA follows the same 3000-update training protocol and four pose conditions, with the same homogeneous detector adaptation as EmbodiedScan (Table 6(c)).

StreamPETR Static predicts one camera-axis depth per image token at each decoder layer and shares the resulting point across its point-encoding heads. Query rays use a physical source camera; queries and keys are expressed in the same camera coordinate system. This configuration was fixed before the table was assembled. Its absolute detection AP remains low. StreamPETR has no corresponding camera/depth-noise evaluation; its perturbed entries remain unavailable. Detection-query denoising during training and ordinary augmentation are separate from camera/depth perturbation evaluation.

UniMatch methods use batch size 8 and the fixed training budget in Table 5. Original-input and noisy evaluations use the same final checkpoint for each method. The URoPE and GTA adaptations are specified in Table 6(c).

Across hosts, Static denotes the ray–point RoPE-only model, using the same checkpoint for all metrics. Full is the declared host-specific updating model, not one identical architecture across tasks.

The main table uses the RE10K transfer setting for NVS. The separate representation study in Appendix D.3.4 uses its own training protocol and pose-noise AUC; its scores are not pooled with the main-table transfer results. Comparisons also distinguish the supplied geometric information, attention transformations, and trainable parameters. A baseline with a modified depth interface or positional operator is reported as an adaptation, not pooled with its original formulation. Static encodings and models with geometric feedback remain separate variants. Missing adaptations remain unfilled, not evidence that a method is inapplicable.

Perturbation comparisons hold the geometric inputs, initialization, supervision, and training budget fixed wherever the intervention permits; remaining differences must be specified. Pose-only evaluation is used where depth is not a perturbed input. Natural-estimate comparisons identify the estimator and input residuals where valid references exist; neither reference geometry nor corruption labels are given at inference. Table 1 measures downstream task performance under noisy geometry, not residual pose/depth error or a baseline-normalized corruption index. Let $M_{h,m}(c)$ be the task score for host h, method m and condition c, and let $\mathcal{T}_{h,m}$ be its declared applicable noise types. We define

$$M_{\text{Noisy}}(h,m) = \frac{1}{|\mathcal{T}_{h,m}|} \sum_{t \in \mathcal{T}_{h,m}} \frac{1}{|\mathcal{C}_t|} \sum_{c \in \mathcal{C}_t} M_{h,m}(c), \tag{6}$$

where $\mathcal{C}_P = \{\text{P1D0}, \ldots, \text{P3D0}\}$, $\mathcal{C}_D = \{\text{P0D1}, \ldots, \text{P0D4}\}$, and $\mathcal{C}_{PD} = \{\text{P1D1}, \ldots, \text{P3D4}\}$ contain 3, 4 and 12 cells, respectively. When all three types apply, each receives one-third weight; this differs from equally averaging all 19 cells. Original is P0D0 and is excluded from Noisy. PSNR and AP retain their higher-is-better direction, while AbsRel remains lower-is-better. There is no cross-task aggregate.

Table 6 collects the applicable type sets, aggregation weights, and baseline adaptations without per-cell superscripts in the main table. P and A denote $\mathcal{T}_{h,m} = \{P\}$ and $\mathcal{T}_{h,m} = \{P, D, PD\}$, respectively.

Preserving the evaluated geometry interfaces. RayRoPE predicts encoding depths (Wu et al., 2026) and shares A conditions and weights with Static and Full within each host. P adapters lack this perturbable input; adding a predictor or depth-to-phase pathway changes the baseline. Applicability concerns encoding inputs, not depth outputs; LVSM retains its all-method A grid. Native-interface summaries complement pose-only comparisons but cannot establish a common-distribution ranking across P and A.

Noisy requires all declared conditions at one endpoint; missing measurements neither change applicability nor permit filling depth/joint scores with D0 copies. The new PETR cohort fixed its aggregation and final checkpoint before noisy training and evaluation. Individual pose/depth/joint means remain separate diagnostics. Perturbation strengths and geometry sources are host-specific. In MVSplat, the pose magnitudes are 0.005/0.01/0.02; encoding-depth perturbations apply only to methods using that input. EmbodiedScan uses its frozen multi-view adaptation of this grid; encoding-depth corruption is not full sensor corruption.

Table 6: Noise applicability, aggregation, and baseline adaptations for Table 1. P denotes pose-only evaluation; A includes pose, depth, and joint perturbations. These sets describe the evaluated host interfaces, not universal properties of the encodings.

(a) Applicable perturbations by method and host

Method	LVSM	MVSplat	PETR	EmbodiedScan	UniMatch
Native	A	P	P	P	P
PRoPE	A	P	P	P	P
URoPE	A	P	P	P	P
RoRE	A	P	P	P	P
RayRoPE	A	A	A	A	A
GTA	A	P	P	P	P
GaussRoPE-static	A	A	A	A	A
GaussRoPE	A	A	A	A	A

(b) Conditions and weights in Noisy

Type	Conditions	Count	P weight	A weight
Pose	P1D0–P3D0	3	1	$1/3$
Depth	P0D1–P0D4	4	N/A	$1/3$
Joint	P1D1–P3D4	12	N/A	$1/3$
Original	P0D0	1	Excluded	Excluded

(c) Host-specific baseline adaptations

Host / method	Implemented variant
UniMatch / URoPE	Fixed depth anchors at 0.1 and 5.05.
UniMatch / GTA	No SO(3) branch; base-100 SO(2) rotations.
PETR, EmbodiedScan / GTA	Reference homogeneous branch with translated object-query frames, without separate SO(3) irreducible representations.

Noisy averages conditions within each applicable type, then averages type means with the weights in (b). P protocols have no perturbed encoding-depth input; N/A is not a missing measurement. A reported score requires every condition in its declared set at the same checkpoint. Perturbation magnitudes and training schedules remain host-specific, as detailed in this section. Table 10 reports the common pose-only comparison.

B.2 Selection, Aggregation, and Repetition

Direct robustness comparisons use shared evaluation scenes and condition sets; MVSplat's common pose-only panel is retained separately from its method-dependent Noisy protocols. Condition-level scores accompany aggregates so that the weighting is explicit. Scene-level resampling quantifies variation across evaluated scenes, conditional on the training seed. Primary scores appear in Table 1; supplementary metric panels are reported separately. NVS and MVSplat average target values within scenes; UniMatch averages pairs within scenes, then weights scenes equally. Detection AP uses the native pooled evaluator rather than a mean of per-scene AP. All models use the prescribed final checkpoint, not the best validation checkpoint. These evaluations compare complete host adaptations, including their geometry sources and augmentation policies.

B.3 Pi3X Mechanism Studies

The comparison in Table 7 uses 144 training spaces (288 windows) and 24 development spaces (48 windows) from ScanNet (Dai et al., 2017). The trained variants share a pretrained trunk, input/noise schedule, seed 20260915, batch size 32 and a 500-update budget. Input G_0, the frozen predictor F_0, and untrained Simple fusion are references, not equally trained baselines. Simple uses the same source and frozen prediction with fixed training-derived uncertainty. Direct uses direct residual updates and differs in its prediction/fusion parameterization; it is a package control, not a pure uncertainty ablation. Mean-phase removes sampled phase perturbations but retains scale prediction, fusion and supervision. No-feedback predicts and supervises intermediate states but supplies G_0 to subsequent geometry encoding and state-content conditioning. It is not the main table's RoPE-only Static model.

Table 7: **Pi3X geometry-prediction controls.** Joint pose/depth perturbations on 24 development spaces; 500 updates per trained arm, followed by untrained references. Lower is better; bold marks the best displayed mean, not significance.

Variant	Depth AbsRel	Rotation (°)	Center (mm)
Full reference (F)	**0.0403**	**0.795**	**28.13**
Direct update	0.0760	1.894	54.24
Mean-phase attention	0.0414	0.831	28.64
No state feedback	0.0428	0.851	32.09
Simple fusion	0.0567	0.960	33.86
Frozen predictor (F_0)	0.0849	0.863	30.79
Input geometry (G_0)	0.0760	1.872	67.80

Depth uses perturbed observed pixels; spaces are weighted equally. One training seed. Controls and fixed-checkpoint intervention: Appendix B.3.

Paired control comparisons. Table 7 and the paired comparisons below average observed/partial inputs under joint perturbations at severity 1, first within each physical space and then equally across spaces. Depth AbsRel uses perturbed observed pixels. Differences are $\Delta = \text{Full} - X$; negative values favor Full. Nominal paired-bootstrap 95% intervals use 10,000 resamples of spaces, not training seeds. Reported paired differences retain their supplied precision; subtracting rounded table means can differ in the last displayed digits.

Paired differences: Full minus control.

Control X	Metric	Δ	95% interval
Mean-phase	AbsRel	-0.00107	$[-0.00313, +0.00100]$
	Rotation (°)	-0.0364	$[-0.0550, -0.0079]$
	Center (mm)	-0.51	$[-1.60, +0.73]$
No state feedback	AbsRel	-0.00252	$[-0.00453, -0.00052]$
	Rotation (°)	-0.0561	$[-0.1100, -0.0022]$
	Center (mm)	-3.96	$[-5.25, -2.56]$
Simple fusion	AbsRel	-0.01643	$[-0.02429, -0.00787]$
	Rotation (°)	-0.165	$[-0.291, +0.037]$
	Center (mm)	-5.73	$[-12.36, +1.42]$
Frozen F_0	AbsRel	-0.04455	$[-0.05634, -0.03224]$
	Rotation (°)	-0.068	$[-0.152, +0.016]$
	Center (mm)	-2.66	$[-5.74, +0.43]$

Joint-perturbation strength. The following three conditions all use partial depth inputs and evaluate AbsRel over *all valid observed depths*. This support differs from the perturbed observed pixels used by Table 7 and its paired comparisons, so their absolute depth errors are not directly interchangeable. Differences remain Full minus Mean-phase, with nominal paired-bootstrap 95% intervals.

Joint-perturbation strength: depth AbsRel.

Condition	Mean-phase AbsRel	Full AbsRel	Δ [95% interval]
No added perturbation	0.033551	0.030721	-0.002830 $[-0.004098, -0.001970]$
Joint perturbation $\times 1$	0.035981	0.033331	-0.002650 $[-0.003931, -0.001856]$
Joint perturbation $\times 2$	0.040549	0.038604	-0.001945 $[-0.003232, -0.001004]$

At $\times 1$, joint perturbation combines view-, patch- and pixel-level log-depth noise of 0.05, 0.08 and 0.02 with 6D-twist pose noise of standard deviation 0.02; translation is scaled by scene size. The $\times 2$ condition uses the same random perturbation structure with all depth and pose amplitudes doubled. The first-view pose is anchored, and partial depth inputs retain approximately half of valid patches in

all three conditions, including the condition without added perturbation. These are separate strength conditions, not two sequential perturbations. All three depth-difference intervals are below zero, supporting lower Full depth error on this evaluation support.

Fixed-checkpoint feedback intervention. Table 3(b) reports a separately recorded evaluation with fixed Full weights and zero optimizer updates. Its Normal row is the reference for this intervention; the intervals are not paired with the trained-scheme means in panel (a). Normal consumes $G_0, \ldots, G_4$, whereas hold-G_0 always consumes G_0. All 36 blocks and five readouts execute and recompute their proposals and scales. Normal has lower displayed depth, rotation, and center errors. Depth AbsRel is 0.0403 for Normal and 0.0422 for hold-G_0, a difference of 0.0019 (approximately 4.5% relative to hold-G_0). No revised paired depth interval is available for these means. With the same sign convention, Normal minus hold-G_0 has nominal 95% intervals of $[-0.1365, -0.0788]$ degrees and $[-5.205, -2.829]$ mm. This intervention quantifies feedback sensitivity at a fixed checkpoint, not the effect of retraining without feedback or removing readouts.

Scene bootstrap intervals do not quantify variation across independent training seeds.

C Additional Implementation Details

We distinguish the common state/fusion rule from host-specific tokenization, attention roles, and supervision. Table 8 summarizes the implemented interfaces; the detailed coordinate and training rules below describe Pi3X unless stated otherwise.

Table 8: Geometry inputs and feedback interfaces. Updated depth and pose carry their error scales. Task heads remain host-specific; shared state semantics do not imply identical attention operators or supervision.

Host	Geometry input	Revised state	Reuse in computation
Pi3X	Available observations; native prediction where missing	Depth, non-gauge pose, uncertainty	Four readouts before later decoder attention slots
LVSM (RE10K transfer)	Camera geometry; predicted depth	Depth; context pose except gauge; fixed target pose	Five gaps in six attention blocks
MVSplat	Posed pair; native RGB depth prediction	Depth and relative pose	Five feature-fusion gaps; corrected poses enter Gaussian decoding
PETR	Camera geometry; 3D object queries	Image-key depth and camera pose	Five decoder gaps; updated image geometry
StreamPETR	Camera geometry; temporal object queries	Image-key depth and camera pose	Five decoder gaps; native temporal memory retained
EmbodiedScan	RGB-D, camera geometry; 3D point queries	Image-key depth and supported camera pose	Two updates between three shared attention passes
UniMatch	Posed pair; native RGB depth prediction	Depth and relative pose	Five feature-fusion gaps; corrected poses enter depth matching

Host adaptation contracts. LVSM, MVSplat and UniMatch rotate Q/K while retaining their native V/O and attention windows where applicable. Pi3X and the detector adapters use value transformation and output restoration as well. LVSM fixes the requested rendering pose and first-camera gauge; freezing its geometry-module parameters during transfer does not disable state updates during a forward pass. MVSplat retains its Gaussian/rendering heads and UniMatch its depth head; their internal depth state is not substituted directly for these task outputs.

Detection queries are 3D references; EmbodiedScan uses 3D point queries. The query's attention role is distinct from the coordinate frame used to express its geometry. In the camera-frame realization, a world-frame object reference is transformed into a selected camera frame to construct its descriptors. Its phase inherits that frame's uncertainty, without an associated measured source-camera error that would cancel it. The image keys carry the observation state: camera pose errors are shared across a view, while local depth errors move endpoints along their rays. Attention associations map post-attention query features back to image keys for the geometric readout, excluding

training-only denoising queries where present. Unsupported keys retain their preceding state. The Gaussian updater revises this image-key state; object reference positions and StreamPETR's ego-aligned memory remain governed by the host.

Initial visual depth comes from each host's depth readout rather than a universal Pi3X predictor. Where the supplied pose is also the proposal anchor, it is not an independent visual pose measurement. Auxiliary losses and available references remain task-specific; box supervision alone does not calibrate depth uncertainty. Appendix B.1 records training differences, including bounded pose proposals for LVSM and the encoding-scale changes for MVSplat/UniMatch. The host comparison evaluates these distinct adaptations, not an ablation with one identical attention operator.

C.1 Depth Coordinates and Patch Representatives

Initialization and proposals. Pi3X fills missing geometry using the frozen native estimate F_0 and retains valid input observations in G_0. Patch features predict F_0-anchored depth proposals; camera-pooled features predict pose proposals and source/proposal scales. Each stage fuses its proposal with the original source means, whose predicted error scales may change. The terminal geometry readout is an instance-specific output, not a requirement on other hosts' task heads. A fixed mean need not have zero uncertainty.

Camera-Z depth and radial distance are different coordinates. For a unit camera-frame ray r_i with positive third component, $z_i = \rho_i (r_i)_z$. When the ray is fixed, $\delta \log z_i = \delta \log \rho_i$. In this realization, pixel rays come from the frozen native local-point prediction, not an assumed calibrated patch-center ray. Valid observed depths rescale points along those pixel rays. Patch geometry is obtained by pooling local points, then taking their norm and normalized direction. Changes in within-patch depth can therefore change the pooled ray; it is recomputed rather than treated as a permanently fixed patch direction.

Depth proposals add a patch-shared log-scale correction to the native pixel geometry. Fusion uses pixel log-Z with a patch-shared gain. The uncertainty target, however, is the log radial-distance residual of the *pooled token*: reference points are formed on the same pixel rays, pooled, and compared to the pooled source or visual proposal. It is not the root-mean-square of pixel residuals. A patch enters this NLL only when its reference pixels are valid. Observed, missing, and mixed patches are grouped separately in the loss reduction.

For each diagonal coordinate, Equation 5 gives $k = P_s/(P_s + P_v)$, $m_f = m_s + k(m_v - m_s)$ and $P_f = 2P_sP_v/(P_s + P_v)$. Thus equal source/proposal covariances are preserved rather than halved. Reusing the fixed source avoids recursively treating reused estimates as independent measurements. The fixed-half rule is a symmetric design choice, not an optimality or exact-posterior claim; learned covariance models are not certified error bounds.

C.2 Pose Chart and Covariance Transport

For each camera, fix the source pose T_0 and scene scale s_0. With rotation coordinates first, define

$$S = \mathrm{diag}(1, 1, 1, s_0, s_0, s_0), \quad x(T) = S^{-1} \operatorname{Log}(T_0^{-1} T)^{\vee}, \quad T_f = T_0 \exp\big((S x_f)^{\wedge}\big). \tag{7}$$

The wedge/vee maps convert a six-vector to/from its Lie-algebra matrix. The source mean is zero and the visual mean is $x(F_0) + \Delta x$ in this chart. Camera zero fixes the gauge. The implementation rejects proposals whose rotation coordinates leave its local chart near angle π.

Let $J_n(x) = S^{-1} J_r(Sx) S$ be the right $\mathrm{SE}(3)$ Jacobian in normalized coordinates. The fused chart covariance P_f is transported as $P_{\tan} = J_n(x_f) P_f J_n(x_f)^{\top}$. Native pose uncertainty is first converted into the same source chart using the inverse Jacobian at $x(F_0)$. The diagonal uncertainty interface receives the majorant

$$D_{aa} = \sum_b |(P_{\tan})_{ab}|, \qquad D_{ab} = 0 \quad (a \neq b), \tag{8}$$

rather than simply deleting correlations. For symmetric $P_{\tan}$, $D - P_{\tan}$ is positive semidefinite: each off-diagonal pair contributes $|(P_{\tan})_{ab}|(v_a - \mathrm{sign}((P_{\tan})_{ab}) v_b)^2$ to its quadratic form. This is a matrix bound on the modeled covariance, not a guarantee that the learned model bounds the true estimation error. Translation scales are restored to physical units before geometric propagation.

C.3 Feedback Schedule and Gradient Routing

The geometry loss supervises depth, rotation and translation. Source and proposal scales receive diagonal Gaussian NLL supervision against training-reference residuals. For a residual r and log scale u,

$$\ell_{\mathrm{NLL}}(r,u) = \tfrac{1}{2}\,\mathrm{sg}(r)^2 e^{-2u} + u, \qquad \mathcal{L} = \mathcal{L}_{\mathrm{geometry}} + \lambda_\sigma \mathcal{L}_{\mathrm{source,visual}}. \tag{9}$$

Here sg stops gradients; the second term aggregates source/proposal NLLs over valid residuals and feedback stages. Fused uncertainty follows Equation 5 with no independent NLL target. These scales fit errors against training references, not separately identified sensor noise. Detaching the residual and scale-branch features prevents reducing NLL by moving the proposal or its visual features.

The Pi3X decoder has 36 blocks and five geometry-attention slots at zero-based block indices $(1, 9, 17, 25, 33)$. They consume G_0, G_1, G_2, G_3, G_4. Four intermediate readouts write states for the next slot. A terminal readout, using the final point/camera decoder features, produces G_5 with uncertainty; there is no sixth attention consuming it. Intermediate readouts are untied LayerNorm-plus-linear heads. Depth uses patch features; pose uses features averaged within each camera. Mean and log-scale adjustments are zero-initialized relative to their fixed anchors and priors. A small current-state content projection also conditions the attention features; it does not introduce an extra repeated-source branch.

Initial error scales are estimated separately by input source from training-reference residuals. Depth sources distinguish observed, missing, and mixed patches; pose sources distinguish observed and predicted poses. For each source, the initial standard deviation is the larger of the root-mean-square residual about zero and a configured minimum error scale. These minima prevent initialization at arbitrarily small uncertainty; they are modeling choices, not calibrated sensor-noise levels or theoretical error bounds. Mixed-depth scales are additionally lower-bounded by the observed- and missing-depth scales. Input masks select the source group at inference. Learned log scales are separately bounded to $[-7, 1]$.

The geometry loss supervises the five output states with equal stage weight; the additional source/proposal NLL coefficient is $\lambda_\sigma = 0.01$. Within each stage the two depth NLLs are averaged, as are the two pose NLLs; depth and pose terms are then added and stages averaged. Losses respect valid reference support and the pose gauge mask.

For NLL, residuals and normalized features are detached before the scale projections, isolating its gradients to those projections. The task branch uses non-detached scale predictions. Local phase sensitivities are detached in this propagation implementation, while predicted scales still influence sampled phases and fusion gains. The sampled attention path is also used during inference, with the same error-sharing rules as in training. Each attention call uses one sampled phase bank, with independent draws across descriptor groups, heads, and rotary pairs; there is no outer average over repeated attention outputs. In this implementation, the random streams are keyed by a supplied seed, a stable input-window identifier, the feedback stage, descriptor group, and error type. Holding these identifiers fixed reuses the base draws; changing geometry or scales still changes the phases.

C.4 Computational Cost: Protocol and Results

We evaluate two aspects of the uncertainty module: the cost saved by factorizing relative phases into token-side rotations, and the execution of these rotations within a fused-attention training pipeline. Figure 6 isolates the first question with matched phases; the NVS measurements below evaluate the second in a complete model.

Why compare pairwise and token-side factors? Both implementations receive the same geometry, sensitivities, and shared error samples, and call the same phase-propagation routine. The pairwise implementation explicitly constructs $\Delta\phi_{ij} = \phi_j - \phi_i$ and its sine/cosine factors for every query–key pair. The token-side implementation stores each token's factors and obtains the same relative rotation through the Q/K dot product. Fixing the phases makes this a direct test of the computation and storage eliminated by factorization.

With C views, P tokens per view, and $N = CP$, explicit factors have shape $[B, C, H, P, N, d_h/2]$. Token-side key factors have shape $[BC, H, N, d_h/2]$, with queries taking the within-view slice.

Q has shape $[BC, H, P, d_h]$ and K/V have shape $[BC, H, N, d_h]$. The resulting operands fit the standard FlashAttention interface (Dao et al., 2022), so its existing kernels perform both forward and backward computation. By comparison, our bilateral analytic-CF consumer applies pair-dependent channel factors while forming logits, using chunked eager operations and autograd recomputation. Fusing that calculation would require a specialized attention kernel and its derivatives. Source-only CF in the same consumer already factors by token and uses ordinary SDPA.

Benchmark setup. We use a synthetic scene with four fixed cameras, an analytic depth surface, and sampling seed 0. Camera-owned pose errors are shared across tokens; token-owned depth errors are reused across Q/K roles. Descriptor groups, heads, and rotary pairs have independent sample streams. The separate-head reference uses $B = 1$, $C = 4$, $H = 16$ (eight ray and eight point heads), and $d_h = 48$. We sweep $P \in \{16, 32, 64, 128, 256, 384, 512, 768, 1024, 2048\}$ on one NVIDIA A100-PCIE-40GB with PyTorch 2.6.0, CUDA 12.4, and FP32. Autocast, TF32, and gradient tracking are disabled during timing. The dense reference materializes its factors with eager, unchunked, out-of-place operations; the token-side arm stores only the per-token factors.

Equivalence check. Before timing, we verify the two representations at $P = 16$ with identical token phase banks. Maximum absolute differences are 7.70×10^{-6} for relative sine/cosine factors, 7.04×10^{-6} for induced logits, and 2.60×10^{-5} for phase gradients. All pass elementwise absolute and relative tolerances of 2×10^{-4}. Reversing the relative-phase sign fails the check, confirming that the test detects an incorrect rotation.

Timing and memory. The timed module starts from prepared sensitivities and sampled physical errors and computes phase propagation and sine/cosine factors. Random draws, uncertainty scaling, upstream Jacobians, feature rotation, attention, readouts, and backward computation lie outside this interval. After five warmup calls per arm, we alternate arm order over 30 paired repetitions and measure synchronized CUDA-event durations. We report median time and maximum additional peak allocation above the common resident inputs. At $P = 512$, timing interquartile ranges are $[2.297, 2.331]$ ms for token-side and $[27.653, 27.795]$ ms for dense preparation. The dense arm runs out of memory at $P = 768, 1024, 2048$, while token-side preparation completes the sweep. Figure 6 marks these measured failures with crosses. Attention itself retains its quadratic arithmetic cost.

Fused-attention execution. We next profile the six-layer NVS model at batch size eight on an A100 40 GB (PyTorch 2.6.0, CUDA 12.4), with Q shaped $[40, 16, 1024, 48]$ and K/V shaped $[40, 16, 5120, 48]$. Using the same checkpoint and input, we compare native FP32 memory-efficient SDPA with BF16 FlashAttention applied only to prepared Q/K/V. Geometry, phase preparation, parameters, and the returned output stay in FP32. After one warmup per arm, one profiled forward/backward pass records six Flash forward and six Flash backward calls, confirming that the token-side path executes through the fused backend. All 98 measured gradient tensors are finite; the BF16 gradients have relative L_2 difference 43.7% and cosine similarity 0.9794 against FP32. We evaluate the resulting training quality in the full run below.

Full-training cost and quality. The NVS backend comparison uses 1,500 optimizer updates, batch size eight, seed 0, identical initialization and ordered inputs, and the same optimizer, schedule, and loss. Both arms retain layerwise pose/depth updates; all 20 sigma parameters remain unchanged. The candidate uses BF16 Flash only at the prepared-Q/K/V boundary during training; both evaluations use FP32. From the first training iteration through the final optimizer update, FP32 SDPA takes 105.622 min and BF16 Flash takes 82.917 min, with peak allocated memory of 34.568 and 31.350 GB (decimal), respectively. This interval covers the training loop, including any checkpoint work. The runs use different physical A100 40 GB cards with different concurrent loads; these are recorded whole-training timings.

On the 17-sequence development panel, the BF16-minus-FP32 noisy PSNR difference is -0.00323 dB (95% paired-sequence bootstrap interval $[-0.06461, 0.05286]$), and the LPIPS difference is $+0.000626$ ($[-0.001859, 0.003531]$). These noisy-input intervals satisfy the prespecified quality criteria: the PSNR interval's lower endpoint is at least -0.1 dB, and the noisy LPIPS interval's upper endpoint is at most 0.01. This complements the factor-preparation benchmark with observed fused execution, whole-training cost, and final image quality.

C.5 Relative-Phase Propagation

Propagation accuracy. We compare the Gaussian relative-phase factor with nonlinear Monte Carlo on a fixed Pi3X Full checkpoint after 500 updates. Four training spaces selected by a fixed hash (scene0705, scene0078, scene0006, scene0597) contribute two previously exposed windows each. Each window has four 224×224 views and 256 tokens per view. Known-scale propagation uses sensor-pinhole patch geometry; predicted-scale propagation uses the frozen G_1 state. These are approximation diagnostics, not held-out task evaluation or uncertainty calibration.

The known distribution anchors camera zero and uses right-local pose errors with per-axis rotation standard deviation 0.02 radians, translation standard deviation 0.02 times the scene scale, and independent patch log-radius standard deviation 0.08. A multiplier λ scales these standard deviations. In the predicted case it scales the predicted distribution itself. The reference exponentiates pose and log-depth draws through the nonlinear geometry. Absolute complex factor error is estimated with 4096 draws and a common-random linear control variate, then averaged equally over spaces. FP64 reference and independent Jacobian checks separate numerical error from linearization error.

Table 9: Relative-phase factor error under joint uncertainty. Space-equal means; parentheses give the mean numerical Monte Carlo standard error, not training-seed variation.

Scale source / λ	0.25	0.5	1	2	4
Known	0.000251	0.000940	0.002926	0.007779	0.018755
MC standard error	(0.000038)	(0.000152)	(0.000604)	(0.002211)	(0.006071)
Predicted	0.000068	0.000250	0.000917	0.003027	0.008415
MC standard error	(0.000015)	(0.000059)	(0.000237)	(0.000918)	(0.002924)

Approximation error grows with scale; finite Monte Carlo error remains material at high noise.

All quantities here are distinct from the main table's host adaptations.

D Derivations and Additional Analyses

D.1 Propagation and Shared-Error Derivations

D.1.1 Local point Gaussians and shared camera errors

Linearizing $X_i = C_{c(i)} + \rho_i D_i$ at the current geometry gives

$$\delta X_i \simeq J_i^{\mathrm{pose}} \delta\xi_{c(i)} + \rho_i D_i \delta\eta_i, \qquad \Sigma_{X_i} \simeq J_i^{\mathrm{pose}} P_{c(i)} (J_i^{\mathrm{pose}})^\top + \rho_i^2 \sigma_{\eta,i}^2 D_i D_i^\top. \tag{10}$$

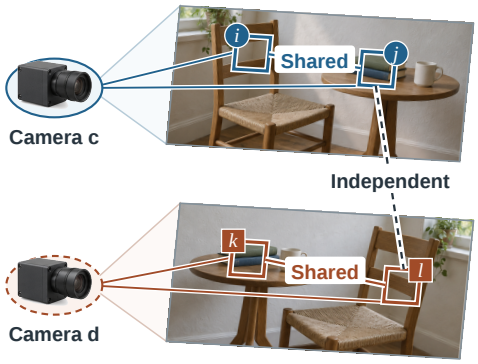
Here J_i^{pose} is the pose-to-point Jacobian. Tokens i, j from the same camera also share covariance $J_i^{\mathrm{pose}} P_c (J_j^{\mathrm{pose}})^\top$. Their marginal covariances alone therefore do not determine the uncertainty of their relationship. Figure 7 isolates shared translation errors with directions and depths fixed. More generally a shared source cancels only when the two phase sensitivities agree; sharing need not reduce every relative uncertainty.

D.1.2 World-frame invariance of the geometry interface

For the image-token realization in Section 3.3, let $A \in \mathrm{SE}(3)$ change the common world frame and let each camera-to-world pose become $T_c' = AT_c$. Keep camera-local rays, radial depths, intrinsics, and the rigid-transform-invariant scene scale s fixed. For source camera k and reference camera c,

$$(T_c')^{-1} T_k' = (AT_c)^{-1}(AT_k) = T_c^{-1} T_k. \tag{11}$$

Thus the reference-frame camera center, direction, and point are unchanged. Their ray moment $C_i^{(c)} \times D_i^{(c)}/s$, point coordinate $X_i^{(c)}/s$, and frequency-projected phases in Equation 2 are unchanged as well. The six-coordinate ray descriptor encodes direction and line moment; the three-coordinate point descriptor encodes the endpoint formed using depth. These descriptor dimensions are distinct from attention-head width. The line alone does not specify a unique camera origin along it, and the combined descriptor need not uniquely recover every geometric variable. For a world-frame 3D query point with homogeneous coordinate Q, the same frame change gives $Q' = AQ$ and



Shared within a camera. Points i, j (or k, l) move together under a camera's translation error; this term cancels from their difference.

Independent across cameras. Points j, l inherit different camera errors, which need not cancel.

For 3D RoPE. Equal pointwise uncertainties can yield different relative-phase uncertainties. Their dependence matters for the encoding.

Figure 7: Shared camera errors. The translation-only example holds directions and depths fixed; token-local depth errors are not illustrated. Synthetic views.

$(T'_c)^{-1}Q' = T_c^{-1}Q$. Expressing this point in a camera frame leaves its role as a 3D point query unchanged.

The same reasoning applies to uncertainty when pose perturbations use right-local coordinates. Write $T_c(\epsilon) = T_c \exp(\delta\xi_c^\wedge)$ and $\rho_i(\epsilon) = \bar{\rho}_i \exp(\delta\eta_i)$. Under the frame change, $T'_c(\epsilon) = AT_c(\epsilon)$, so Equation 11 holds for every realization of the same local errors. The phase functions therefore agree as functions of those errors, not only at their means. Differentiating at zero gives identical phase sensitivities $a_{i,o}$; keeping the same local covariance P_o preserves v_{ij} and the Gaussian phase expectation. With the same samples for each camera and token-depth error source, the sampled rotary factors are identical. For fixed input features, the geometry-conditioned attention operator is therefore unchanged.

This result concerns a common coordinate change, not independent errors in estimated camera poses. It assumes local covariances and sampling sharing follow the stated coordinate convention. It does not prove that absolute-coordinate features, a host's prediction head, or every learned update rule are invariant. Whole-model invariance would require checking those additional paths separately.

D.1.3 Why shared error sources determine relative uncertainty

For the linearized phase model, define $\kappa_{ij} = \mathrm{Cov}(\phi_i, \phi_j)$ within a fixed query frame, channel family, head, and rotary pair. Independence between distinct error sources (cameras or token depths) gives

$$\kappa_{ij} = \sum_o a_{i,o}^\top P_o a_{j,o}, \qquad v_{ij} = \kappa_{ii} + \kappa_{jj} - 2\kappa_{ij}. \tag{12}$$

Marginal variances alone omit κ_{ij}. If an error source's coefficients are equal for two tokens, its contribution to their relative variance is zero. In particular, using the same perturbed camera in both roles makes $T_c(\epsilon)^{-1}T_c(\epsilon) = I$; assigning independent source and query-frame samples would break that cancellation. Distinct token-depth errors generally remain. The coefficients also depend on the current geometry, so a layerwise mean update requires refreshing the propagation Jacobians.

These statements are exact for the stated linear Gaussian phase model. Nonlinear geometry is represented by a local approximation; large pose or depth errors can invalidate its accuracy. The model also does not retain arbitrary correlations between different cameras.

Ray–point descriptor Jacobians. For source camera $k = c(i)$ and query camera c, abbreviate $A = R_c^\top R_k$, $\beta = R_c^\top(C_k - C_c)$, $d = Ar_i$, and $x = \beta + \rho_i d$. Let $[z]_\times v = z \times v$. Under the right-local perturbations above, write $\delta\xi_o = (\delta\theta_o, \delta t_o)$ with translation in physical units. Holding the ray and scene scale fixed, the first-order variations are

$$\begin{aligned} \delta d &= -A[r_i]_\times \delta\theta_k + [d]_\times \delta\theta_c, \\ \delta\beta &= A\delta t_k + [\beta]_\times \delta\theta_c - \delta t_c, \\ \delta x &= \delta\beta + \rho_i \delta d + \rho_i d\, \delta\eta_i. \end{aligned} \tag{13}$$

Consequently, the ray and point descriptor differentials are

$$\delta g^{(c)}_{i,\mathrm{ray}} = \begin{bmatrix} \delta d \\ (-[d]_\times \delta\beta + [\beta]_\times \delta d)/s \end{bmatrix}, \qquad \delta g^{(c)}_{i,\mathrm{point}} = \delta x/s. \tag{14}$$

Collecting the coefficient of each physical error source gives $J_{g,i,o}$ and $a_{i,o} = J^\top_{g,i,o}\omega_{h,b}$. When $k = c$, the two pose roles use the same perturbation and their coefficients are added before forming the covariance. With $A = I$ and $\beta = 0$, both pose contributions cancel; token-depth variation remains in the point descriptor. Translation scales in the normalized fusion chart are restored as specified in Appendix C.2.

D.2 Attention Approximations and Refinement Details

The Gaussian characteristic function gives the expected relative rotary factor, but softmax is nonlinear. Applying softmax to expected logits and averaging sampled attention are different approximations; neither equals the other in general. Our implementation samples camera and token-depth errors, forms token-side rotations, and applies Equation 4. Error-source sharing is preserved within the specified descriptor-family/head/rotary-pair indices. Independent random streams across these indices need not describe one jointly sampled 3D scene.

Avoiding pairwise expansion follows from the rotary factorization

$$R(\phi_i)^\top R(\phi_j) = R(\phi_j - \phi_i), \tag{15}$$

with the same error samples used on both sides. Linearization makes phase propagation tractable; it is this factorization and sample reuse that preserve a standard attention interface. Q/K/V rotation and inverse query rotation of the output remain part of the operator. Analytically evaluating the characteristic function for each query–key pair instead gives pair-dependent amplitudes and expected logits. This can be chunked, so pairwise dependence does not imply simultaneous allocation of a dense tensor. Conversely, some source-only characteristic-function approximations already factor by token. Appendix C.4 therefore isolates representation cost under the same sampled model, rather than attributing differences between analytic expectations and sampling solely to implementation.

D.3 Additional Experimental Results

D.3.1 Complete task and perturbation results

This section expands Table 1 with supporting metrics and perturbation comparisons. For image synthesis, we report PSNR, structural similarity (SSIM) (Wang et al., 2004), and learned perceptual image patch similarity (LPIPS) (Zhang et al., 2018). Pose, depth, and joint perturbations are included only where the task protocol supports them. Different training policies and natural geometry sources remain separate comparisons. Table 10 reports supplementary task metrics and pose-only evaluations; primary Original scores are in Table 1. The pose-only panel uses common perturbation conditions within each host, distinct from the three-type Noisy score in Equation 6. In Table 10, Full improves all reported pose-only metrics over Static. On Original inputs, it improves all reported metrics except NVS PSNR: the change from 23.48 to 23.45 dB accompanies higher SSIM (0.647 to 0.690) and lower LPIPS (0.385 to 0.288), rather than a uniform image-quality decline. Table 11 retains StreamPETR. Main-table AP_{25} uses TTA over all 67 seen classes; the no-TTA results in Table 10 are separate diagnostics. From Static to Full, the current MVSplat grid gives a joint-perturbation PSNR mean of approximately 17.94 dB versus 18.28 dB, and a depth-only mean of 18.46 dB versus 18.60 dB, computed from the displayed values in Table 12. For UniMatch under the common pose-only conditions, Full improves on Static but not RayRoPE: Full's AbsRel/RMSE/δ_1 are 1.871/3.259/12.90%, versus RayRoPE's 1.852/3.029/14.61% (Table 10(b)). The joint-only AbsRel entries are 1.809 for Full and 1.848 for RayRoPE in Table 17. The main-table Full–RayRoPE comparison uses the same three-type conditions and weights for both methods (Table 6), not a pose-only score for one and a three-type score for the other. The reported 7.4% reduction is specific to that aggregate; it does not assert superiority on pose-only inputs or every perturbation type. Pose-only evaluation probes camera errors but leaves encoding-depth errors untested. For the pose-only diagnostic, EmbodiedScan Full's TTA AP_{25} is 12.23%, compared with 11.96% for PRoPE. Full's main-table Noisy score is 12.43%: it averages pose, depth and joint type means of

Table 10: Supplementary task evaluations with matched metric rows. Panel (a) includes the Original primary scores from Table 1; panel (b) averages pose-only conditions P1D0–P3D0, distinct from three-type Noisy. Supplementary endpoints are interpreted separately. AP, NDS and δ_1 are percentages; T denotes TTA. Bold/underline: best/second-best displayed value. Baseline adaptations: Table 6.

(a) Original inputs

Task / dataset	Metric	Native	PRoPE	URoPE	RoRE	RayRoPE	GTA	GaussRoPE	
								Static	Full
NVS	PSNR ↑	17.24	22.46	**24.36**	22.68	22.11	22.49	<u>23.48</u>	23.45
RE10K	SSIM ↑	0.504	0.656	**0.729**	0.654	0.626	0.656	0.647	<u>0.690</u>
	LPIPS ↓	0.670	0.330	**0.234**	0.355	0.397	0.332	0.385	<u>0.288</u>
MVSplat	PSNR ↑	18.29	18.28	18.13	18.16	18.21	17.66	<u>18.43</u>	**18.59**
RE10K	SSIM ↑	**0.627**	0.614	0.614	0.614	0.621	0.605	0.614	<u>0.624</u>
	LPIPS ↓	0.342	0.335	0.342	0.350	<u>0.334</u>	0.349	0.339	**0.333**
PETR	mAP ↑	1.24	0.51	1.02	3.70	4.05	0.48	<u>5.15</u>	**5.18**
nuScenes	NDS ↑	7.58	4.96	5.20	10.19	11.32	5.17	<u>11.77</u>	**11.82**
EmbodiedScan	AP_{25}^{T} ↑	11.71	<u>12.39</u>	10.74	11.15	12.01	12.24	11.46	**12.81**
ScanNet60	AP_{25} ↑	12.02	12.65	11.21	11.74	10.42	<u>13.46</u>	13.07	**13.78**
UniMatch	AbsRel ↓	1.025	1.201	<u>0.952</u>	0.985	**0.805**	1.245	1.008	0.963
SUN3D	RMSE ↓	2.113	2.237	<u>1.810</u>	1.996	**1.574**	2.316	2.402	2.004
	δ_1 ↑	21.62	18.85	<u>22.66</u>	21.74	**23.80**	20.10	14.65	18.79

(b) Pose-perturbed inputs

Task / dataset	Metric	Native	PRoPE	URoPE	RoRE	RayRoPE	GTA	GaussRoPE	
								Static	Full
NVS	PSNR ↑	17.21	22.42	**23.17**	21.98	21.55	22.44	21.84	<u>22.61</u>
RE10K	SSIM ↑	0.503	0.655	**0.694**	0.636	0.613	0.655	0.631	<u>0.667</u>
	LPIPS ↓	0.671	0.331	**0.250**	0.364	0.404	0.332	0.397	<u>0.303</u>
MVSplat	PSNR ↑	17.62	17.74	17.70	17.80	17.76	17.26	<u>17.92</u>	**18.28**
RE10K	SSIM ↑	0.576	0.585	0.588	0.584	<u>0.591</u>	0.582	0.590	**0.598**
	LPIPS ↓	0.366	0.349	0.356	0.365	<u>0.348</u>	0.362	0.355	**0.344**
PETR	mAP ↑	1.23	0.51	1.02	3.65	4.01	0.48	<u>5.10</u>	**5.13**
nuScenes	NDS ↑	7.55	4.93	5.20	10.15	11.30	5.15	<u>11.75</u>	**11.76**
EmbodiedScan	AP_{25}^{T} ↑	11.51	<u>11.96</u>	10.63	11.13	11.84	11.67	11.35	**12.23**
ScanNet60	AP_{25} ↑	11.85	12.57	10.76	11.95	10.12	**12.64**	11.72	<u>12.63</u>
UniMatch	AbsRel ↓	2.520	2.672	2.108	2.208	**1.852**	2.442	2.233	<u>1.871</u>
SUN3D	RMSE ↓	4.216	4.262	3.587	3.638	**3.029**	3.905	3.758	<u>3.259</u>
	δ_1 ↑	11.99	12.80	**14.78**	14.12	<u>14.61</u>	13.38	11.82	12.90

Table 11: Supplementary detection evaluations. mAP and NDS are percentages. StreamPETR has original-input evaluation only. These evaluations are interpreted separately from the primary endpoints. Bold marks the best displayed point estimate; adaptations are in Table 6.

Task / dataset	Metric	Native	PRoPE	URoPE	RoRE	RayRoPE	GTA	GaussRoPE	
								Static	Full
StreamPETR	mAP ↑	0.0013	0.0040	0.0406	0.0055	**0.4766**	0.0137	0.0457	0.1635
nuScenes	NDS ↑	1.48	1.50	1.90	1.57	**2.33**	1.64	1.80	1.87

12.229%, 12.820% and 12.227%, respectively. PRoPE uses only pose noise because encoding-depth perturbations are inapplicable.

In the mixed-training main table, Full has higher AP_{25} than Static on both input conditions: 12.81 versus 11.46 original, and 12.43 versus 11.38 Noisy, with TTA. All main-table methods execute three attention passes; Full updates geometry between them. These comparisons describe each complete training and architecture configuration.

D.3.2 MVSplat response across the perturbation grid

Figure 5 uses all eight MVSplat endpoints from Table 1 for pose perturbations; RayRoPE, Static and Full additionally have applicable depth/joint conditions. Every cell averages the same 38 RE10K development scenes, with three target images per scene, seed 111124 and 2500 training updates. The figure displays all pose-only and depth-only cells and the Full–Static difference in all 12 joint cells; Table 12 gives the absolute scores. These are complete host adaptations, including differences in geometry modules and auxiliary losses, not a single-component ablation.

P0–P3 use rotation scales $0, 0.005, 0.01, 0.02$ radians and translation scales multiplied by the context-camera baseline. The first camera is fixed; the second receives sequence bias and frame jitter at 0.1 times the bias scale. D0–D4 use log-depth bias, jitter standard deviation and missing fraction (b, σ, r) of $(0, 0, 0)$, $(0, 0.01, 0)$, $(0.025, 0.01, 0.05)$, $(0.05, 0.025, 0.1)$ and $(0.1, 0.05, 0.2)$. Depth corruption acts on a connected quarter-region of the non-anchor view's encoding depth; r is relative to that region. D labels are ordered conditions combining three changes, not a single physical scale. RGB targets and evaluation references remain fixed. Curves show absolute PSNR with common limits and no smoothing; no training-seed error bars are inferred from this single-seed comparison.

Table 12: MVSplat PSNR (dB) across the complete evaluated grid. N/A denotes inapplicable encoding-depth perturbations. Static and Full denote the two GaussRoPE variants.

Pose	Depth	Native	PRoPE	URoPE	RoRE	RayRoPE	GTA	Static	Full
P0	D0	18.29	18.28	18.13	18.16	18.21	17.66	18.43	**18.59**
P0	D1	N/A	N/A	N/A	N/A	18.21	N/A	18.43	**18.60**
P0	D2	N/A	N/A	N/A	N/A	18.20	N/A	18.45	**18.60**
P0	D3	N/A	N/A	N/A	N/A	18.19	N/A	18.47	**18.60**
P0	D4	N/A	N/A	N/A	N/A	18.18	N/A	18.49	**18.60**
P1	D0	18.03	18.03	17.96	18.02	18.03	17.50	18.23	**18.53**
P1	D1	N/A	N/A	N/A	N/A	18.03	N/A	18.23	**18.53**
P1	D2	N/A	N/A	N/A	N/A	18.02	N/A	18.24	**18.52**
P1	D3	N/A	N/A	N/A	N/A	18.02	N/A	18.26	**18.53**
P1	D4	N/A	N/A	N/A	N/A	18.01	N/A	18.28	**18.52**
P2	D0	17.69	17.79	17.76	17.84	17.81	17.31	17.99	**18.34**
P2	D1	N/A	N/A	N/A	N/A	17.81	N/A	17.98	**18.34**
P2	D2	N/A	N/A	N/A	N/A	17.81	N/A	18.00	**18.34**
P2	D3	N/A	N/A	N/A	N/A	17.80	N/A	18.01	**18.34**
P2	D4	N/A	N/A	N/A	N/A	17.80	N/A	18.04	**18.34**
P3	D0	17.15	17.41	17.40	17.54	17.45	16.98	17.55	**17.98**
P3	D1	N/A	N/A	N/A	N/A	17.45	N/A	17.55	**17.98**
P3	D2	N/A	N/A	N/A	N/A	17.46	N/A	17.58	**17.98**
P3	D3	N/A	N/A	N/A	N/A	17.46	N/A	17.57	**17.99**
P3	D4	N/A	N/A	N/A	N/A	17.46	N/A	17.58	**17.99**

D.3.3 Qualitative display protocol

Figure 4 compares direct NVS, Gaussian reconstruction, 3D detection, and depth estimation under P3D0 pose perturbations, with no added depth noise. Static denotes RoPE-only GaussRoPE; Full includes propagation and updates. Each row shares inputs, target views and crops, and Scene locates the common region. RGB outputs use fixed checkpoints; PSNR is measured on the displayed PNG crops rather than complete benchmark images.

Detection uses saved predictions with score ≥ 0.10; its numbers are the best same-class 9DoF IoU for the selected door (blue) and toilet (orange), respectively. Predictions are solid and GT is dashed. Depth maps share a logarithmic 0.1–10 m scale (yellow near, purple far); invalid GT is gray. AbsRel uses the common valid pixels.



Figure 8: Additional multi-task qualitative examples. Each row specifies its input condition; D0 adds no depth noise. Numbers are displayed-image/crop PSNR, selected-bin IoU, or valid-pixel AbsRel, not benchmark averages. Orange solid/blue dashed boxes denote prediction/GT; gray depth pixels are invalid.

D.3.4 RAY AND DEPTH-DEPENDENT ENCODINGS FOR VIEW SYNTHESIS

Controlled representation comparison. Table 13 compares ray-only, ray–point, and segment-bound encodings in a shared six-layer LVSM with 16 heads. Each arm is trained for 5,000 updates on 39 RE10K scenes with seeds 0, 1, and 2, using matched initialization, data order, and pose-noise schedules. Evaluation uses four context views and three targets on 38 scenes. Scores are normalized trapezoidal AUCs over seven coupled rotation/translation noise levels from 0 to 0.03 in increments of 0.005; they differ from the Noisy aggregation in Table 1.

Ray uses 16 Plücker-ray heads; ray–point uses eight ray and eight predicted-depth point heads; segment bounds uses six direction, six moment, and two heads for each endpoint. Wavelength ranges are $[0.1, 4]$ for ray and ray–point and $[0.1, 2]$ for segment bounds. Sampling is switched off or uses one shared-error sample with supplied pose-noise scales and predicted depth widths. The paired arms retain the same geometry and frequencies. This separate-head component study complements the main transfer comparison: it examines representation choices rather than the full within-head GaussRoPE implementation.

Table 13: **NVS encoding controls.** RE10K, 5,000 updates, three seeds, 38 evaluation scenes. Normalized pose-noise AUC (mean ± seed SD). Sampled adds shared-error sampling in separate-head component variants; protocol: Appendix D.3.4.

Encoding	Uncertainty	PSNR↑	SSIM↑	LPIPS↓
Ray	Off	21.868±0.023	0.6745±0.0011	0.2189±0.0003
Ray–Point	Off	21.782±0.101	0.6694±0.0037	0.2277±0.0020
Segment bounds	Off	21.683±0.050	0.6694±0.0028	0.2219±0.0019
Ray	Sampled	22.074±0.052	0.6798±0.0011	0.2318±0.0016
Ray–Point	Sampled	21.984±0.182	0.6757±0.0050	0.2382±0.0016
Segment bounds	Sampled	21.872±0.121	0.6749±0.0045	0.2342±0.0067

Task-dependent geometry use. Ray-only has the best mean AUC on all three metrics among these three geometric representations, with sampling both off and on. Its PSNR advantage over ray–point is approximately 0.09 dB in both cases. These results suggest that rays and visual features already supply useful constraints for synthesis, while point and segment-bound phases additionally depend on estimated depths. Head allocations and frequency ranges also differ across families. The paired sampling controls isolate uncertainty use within each fixed representation. Sampling improves PSNR and SSIM AUC in all three representations, with higher LPIPS. For ray–point, PSNR changes from 21.782 to 21.984 dB and LPIPS from 0.2277 to 0.2382.

D.3.5 PERTURBATION RESPONSES ACROSS ADDITIONAL TASKS

Tables 14–17 compare eight fixed-checkpoint methods. P0–P3 use D0; Depth/Joint average four/twelve conditions. N/A denotes inapplicability, not missing or zero scores. Arrows indicate preferred directions; bold and underlining mark the best and second-best displayed values in each column, respectively. Protocols follow Table 5 and Appendix B.1: TTA for EmbodiedScan, scene-equal aggregation for synthesis/depth, and native pooling for detection. P1–P3 scales are host-specific. Single-seed comparisons isolate neither components nor training-seed variability.

Full has better P1–P3 scores than Static in all hosts, consistent with Table 1's Noisy comparison; other rankings vary by host/severity.

Table 14: LVSM / RE10K: PSNR (dB)↑ under geometric perturbations.

Method	P0	P1	P2	P3	Depth	Joint
Native	17.238	17.233	17.221	17.182	17.238	17.212
PRoPE	22.462	22.456	22.437	22.366	22.462	22.420
URoPE	**24.355**	**23.954**	**23.307**	22.247	**24.355**	**23.169**
RoRE	22.682	22.500	22.119	21.320	22.682	21.980
RayRoPE	22.109	21.975	21.679	20.993	22.084	21.537
GTA	22.489	22.481	22.460	**22.387**	22.489	22.443
GaussRoPE-static	23.480	22.323	21.963	21.234	22.490	21.850
GaussRoPE	23.446	23.228	22.756	21.851	23.446	22.612

Table 15: PETR / nuScenes: mAP (%)↑ under geometric perturbations.

Method	P0	P1	P2	P3	Depth	Joint
Native	1.244	1.241	1.237	1.223	N/A	N/A
PRoPE	0.512	0.512	0.512	0.508	N/A	N/A
URoPE	1.025	1.019	1.020	1.023	N/A	N/A
RoRE	3.701	3.693	3.671	3.600	N/A	N/A
RayRoPE	4.050	4.034	4.026	3.973	4.048	4.009
GTA	0.477	0.476	0.476	0.475	N/A	N/A
GaussRoPE-static	5.151	5.144	5.122	5.039	5.152	5.099
GaussRoPE	**5.184**	**5.162**	**5.153**	**5.077**	**5.180**	**5.130**

Table 16: EmbodiedScan / ScanNet: AP_{25} (%, TTA)↑ under geometric perturbations.

Method	P0	P1	P2	P3	Depth	Joint
Native	11.711	11.491	11.587	11.462	N/A	N/A
PRoPE	12.391	12.292	11.606	11.990	N/A	N/A
URoPE	10.744	10.906	10.637	10.332	N/A	N/A
RoRE	11.149	11.415	11.097	10.870	N/A	N/A
RayRoPE	12.006	12.049	11.827	11.643	12.011	11.844
GTA	12.235	11.616	11.208	**12.182**	N/A	N/A
GaussRoPE-static	11.459	11.529	11.231	11.284	11.457	11.323
GaussRoPE	**12.811**	**12.568**	**12.582**	11.537	**12.820**	**12.227**

Table 17: UniMatch / SUN3D: AbsRel↓ under geometric perturbations.

Method	P0	P1	P2	P3	Depth	Joint
Native	1.025	1.767	2.661	3.132	N/A	N/A
PRoPE	1.201	2.138	2.914	2.965	N/A	N/A
URoPE	0.952	1.443	2.156	2.726	N/A	N/A
RoRE	0.985	1.508	2.218	2.898	N/A	N/A
RayRoPE	**0.805**	**1.093**	1.894	2.570	0.833	1.848
GTA	1.245	1.743	2.463	3.119	N/A	N/A
GaussRoPE-static	1.008	1.864	2.288	2.547	1.297	1.915
GaussRoPE	0.963	1.405	**1.861**	**2.346**	**0.517**	**1.809**

E SHARED-CAMERA ERROR ABLATION

Intervention. S shares each physical camera-pose error draw across its source tokens and corresponding query coordinate system. D draws source-pose errors independently per token and query-coordinate errors independently of all sources, retaining sharing within each query coordinate system. Source draws are reused across query coordinate systems. This removes source-camera sharing and its common-mode cancellation, not all phase correlations. Self Q/K phases remain identical; depth draws, geometry, frequencies and uncertainty parameterization follow the same rules. No sigma multiplier is used. Equal translation sensitivities cancel common translation in S; general ray/rotation sensitivities need not cancel fully.

Matched training and evaluation. Both variants share full-model initialization (seed 111124), initial priors, training frames and perturbed inputs. Native Q/K-only attention, V/O, windows, cost volume, Gaussian decoder and renderer are retained. All 12 attention consumers in six blocks use the intervention; five geometry updates remain. Training uses 39 RE10K scenes, 256×256 inputs, two context/four target views, batch two with accumulation two, and 2500 Adam updates (10,000 exposures). The learning rate is 2×10^{-4} with OneCycle; RGB, pose-mean and uncertainty losses have weights 1, 0.1 and 0.01. The depth prior uses log-uniform near/far support, not ground-truth calibration. Weights and scales learn separately: this tests adaptation, not a fixed-weight operator toggle.

The fixed final endpoint uses 38 scenes, three targets each and 20 conditions, with 760 rows paired by scene, targets and input/reference identity per variant. Scores average targets, conditions, then scenes. Prespecified *Perturbed-19* averages all noisy conditions: pose/depth/joint weights are $3/19$, $4/19$, $12/19$, unlike Table 1's equal-type Noisy. These new endpoints do not replace the historical main-table models.

Results and scope. Table 18 uses 10,000 paired-scene bootstrap resamples (seed 111124). All three Perturbed-19 intervals favor S; original PSNR/SSIM intervals include zero. These intervals describe reused-scene variation for one training pair, not across-seed uncertainty. These results support camera sharing under this MVSplat protocol; they do not establish gains across all hosts.

Table 18: **Shared-camera ablation by perturbation type.** S retains camera sharing; D removes source-camera sharing. Intervals use paired scene bootstrap; positive S−D favors S for PSNR/SSIM, negative for LPIPS.

Condition	Metric	S	D	S−D	95% interval
Original	PSNR	18.4017	18.1444	+0.2572	[-0.0116, +0.5386]
	SSIM	0.6056	0.6030	+0.0025	[-0.0092, +0.0140]
	LPIPS	0.3446	0.3804	-0.0358	[-0.0511, -0.0229]
Perturbed-19	PSNR	18.0840	17.7652	+0.3188	[+0.0580, +0.5961]
	SSIM	0.5900	0.5796	+0.0105	[+0.0010, +0.0202]
	LPIPS	0.3558	0.3936	-0.0378	[-0.0525, -0.0254]
Pose only	PSNR	18.0017	17.6614	+0.3404	[+0.0774, +0.6199]
	SSIM	0.5860	0.5732	+0.0128	[+0.0034, +0.0227]
	LPIPS	0.3589	0.3970	-0.0380	[-0.0527, -0.0257]
Depth only	PSNR	18.3980	18.1483	+0.2497	[-0.0170, +0.5277]
	SSIM	0.6055	0.6033	+0.0022	[-0.0094, +0.0136]
	LPIPS	0.3444	0.3806	-0.0361	[-0.0515, -0.0233]
Joint	PSNR	17.9999	17.6634	+0.3364	[+0.0746, +0.6144]
	SSIM	0.5859	0.5733	+0.0126	[+0.0032, +0.0226]
	LPIPS	0.3589	0.3971	-0.0382	[-0.0530, -0.0258]