FORMFLOW-GS: A GENERAL AND EFFICIENT METHOD FOR ONLINE MULTIMODAL GAUSSIAN MAPPING

Anonymous authors
Paper under double-blind review

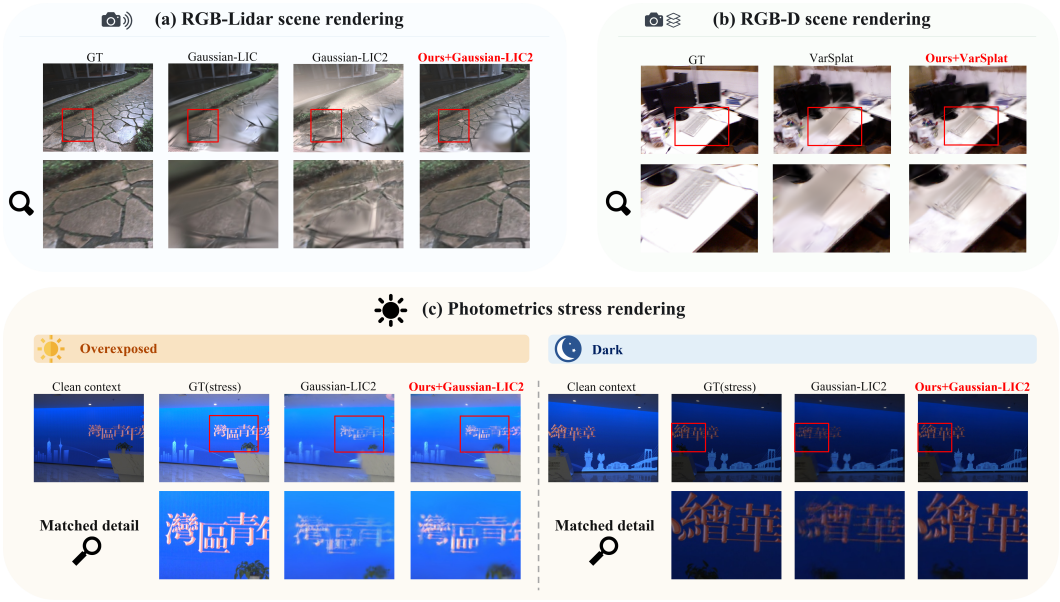

Figure 1: **Qualitative rendering across sensing configurations and photometric conditions.** (a) RGB–LiDAR scene rendering. (b) RGB-D scene rendering. (c) Rendering under overexposed and dark illumination.

ABSTRACT

Multimodal online Gaussian mapping still suffers from critical limitations in geometric formation and refinement efficiency. Although depth or LiDAR provides metric surface geometry, this information is often underused when new Gaussians are formed, while downstream refinement incurs substantial execution overhead. Addressing both stages is therefore essential for high-quality and efficient online mapping. We propose FormFlow-GS, a general Gaussian mapping method built around two components. GeoBirth uses local metric geometry and neighborhood reliability to form surface-aligned Gaussians with adaptive anisotropic support, providing a better-conditioned geometric starting point for refinement. Coalesced Gaussian Optimization (CGO) coalesces heterogeneous attribute updates over a shared visible-Gaussian workset, reducing redundant execution during refinement. The design is broadly applicable and can be readily integrated into Gaussian mapping backbones spanning monocular, RGB-D, RGB–LiDAR, and LiDAR–RGB–IMU settings. Across these backbones and multimodal benchmarks, it consistently improves rendering quality and online mapping efficiency, achieving a strong quality–efficiency balance and outperforming most evaluated state-of-the-art baselines overall.

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.

1 INTRODUCTION

Three-dimensional Gaussian Splatting (3DGS) has become a compelling scene representation for dense SLAM, combining explicit primitives with differentiable rasterization, online map editing, and high-fidelity novel-view rendering (Kerbl et al., 2023). SplaTAM, GS-SLAM, and Gaussian Splatting SLAM (MonoGS) established this paradigm for RGB-D and monocular input (Keetha et al., 2024; Yan et al., 2024; Matsuki et al., 2024). Multimodal systems such as MM3DGS-SLAM, Gaussian-LIC, Gaussian-LIC2, and GS-LIVM further incorporate metric depth, LiDAR, and inertial measurements into online Gaussian mapping (Sun et al., 2024; Lang et al., 2025; 2026; Xie et al., 2025). These systems must continuously form new Gaussians and refine visible ones under a tight online computation budget.

In multimodal Gaussian mapping, RGB provides dense and informative appearance cues but does not directly encode metric 3D surface structure, whereas depth or LiDAR provides sparser but explicit metric constraints on surface location and shape. Such geometric cues are comparatively stable under illumination changes, providing a more robust basis for Gaussian formation. One central question, therefore, is how geometry should influence the formation of new Gaussians. Recent work has improved Gaussian initialization and geometric modeling from different directions: SGAD-SLAM adjusts Gaussian depth along viewing rays, GS-LIVM derives scale and rotation from local covariance, RMGS-SLAM incorporates voxel-PCA structure, and GeoGS-SLAM uses learned geometric priors for map expansion (Hu & Han, 2026; Xie et al., 2025; Li et al., 2026; Gao et al., 2026). Surface-oriented representations such as 2DGS and PGSR further demonstrate the benefit of explicitly encoding surface structure (Huang et al., 2024; Chen et al., 2025a). However, metric geometry remains underused as a reliability-aware formation cue that jointly determines the orientation and anisotropic support of a newborn Gaussian. When newborn Gaussians do not adequately reflect the observed surface geometry, they enter subsequent optimization with geometric mismatch, which can degrade rendering quality and require additional correction.

This additional correction has a direct systems consequence: it increases the visible-Gaussian workload that must be processed by the mapping backend. Recent work has improved Gaussian optimization efficiency through feed-forward prediction, optimizer redesign, and density control (Zhang et al., 2026; Ding et al., 2026; Zhou et al., 2026). Nevertheless, conventional online mappers still commonly update position, appearance, opacity, scale, and rotation through separate attribute-wise paths, repeatedly traversing the same visible Gaussian set and incurring fragmented execution overhead. The two issues are therefore coupled through the mapping loop: Gaussian formation determines both representation quality and the amount of subsequent correction, while backend execution determines the cost of processing that workload.

To address these challenges, we propose **FormFlow-GS**, which couples geometry-aware Gaussian formation with efficient refinement execution. **GeoBirth** uses reliable local LiDAR or depth structure to reduce geometric mismatch at birth, so that newly formed Gaussians require less correction during subsequent refinement. **CGO** then coalesces the remaining visible-Gaussian updates into a shared execution path, reducing repeated scheduling and state access.In this way, FormFlow-GS acts on both sides of the same mapping process: GeoBirth reduces how much refinement is needed, while CGO reduces the cost of executing it, improving reconstruction quality and online mapping efficiency.

As summarized in Fig. 1, our method delivers consistent rendering gains across sensing configurations and transfer settings, while preserving finer visual structure under challenging illumination.

Our contributions are threefold:

- We introduce GeoBirth, which aligns newborn Gaussians with valid local surface geometry and adapts anisotropic support to neighborhood reliability.
- We present CGO, which restructures Gaussian refinement around a shared visibility-aligned workset for coordinated multi-attribute optimization.
- Extensive 2×2, cross-backbone transfer, and novel-view rendering evaluations demonstrate improved reconstruction quality and online efficiency,broad transferability, and strong rendering under challenging illumination.

2 Related Work

Online Gaussian SLAM and multimodal mapping. Online Gaussian mapping has expanded from RGB-D and monocular SLAM to semantic, dynamic, and multimodal settings (Keetha et al., 2024; Yan et al., 2024; Matsuki et al., 2024; Yugay et al., 2023; Li et al., 2024a; Matsuki et al., 2025; Sun et al., 2024; Lang et al., 2025; 2026; Xie et al., 2025). Recent systems further address global consistency, uncertainty, depth-aware placement, and multi-sensor fusion (Zhu et al., 2025; Tran & Kosecka, 2026; Hu & Han, 2026; Usuelli et al., 2026; Li et al., 2026). As increasingly rich sensing provides metric surface structure directly during online mapping, an important question is how this geometry should influence the formation of new Gaussians.

Geometry-aware Gaussian formation. Geometry has been incorporated through surface alignment, planar parameterizations, depth/normal priors, and local point-cloud structure (Guédon & Lepetit, 2024; Huang et al., 2024; Chen et al., 2025a; Turkulainen et al., 2025; Li et al., 2024b). Recent work further exploits surface, depth, and structural cues for geometric reconstruction (Jiang et al., 2025; Wu et al., 2025; Xu et al., 2025; Guédon et al., 2025; Ruan et al., 2025). Online systems also use local covariance, depth adjustment, voxel-PCA, or learned priors to shape new Gaussians (Xie et al., 2025; Hu & Han, 2026; Li et al., 2026; Gao et al., 2026). However, online formation often uses local geometry without explicitly separating validity from reliability. GeoBirth separates these roles: validity determines whether to establish a surface frame, while neighborhood reliability controls the strength of anisotropic shaping.

Efficient Gaussian optimization and execution. 3DGS efficiency has been improved through compression, rendering acceleration, and optimization simplification (Mallick et al., 2024; Lu et al., 2024; Niedermayr et al., 2024; Girish et al., 2024; Fan et al., 2024; Fang & Wang, 2024; Navaneet et al., 2024). Other work improves backend execution, scalability, and optimization efficiency (Ye et al., 2025; Zhao et al., 2025; Hou et al., 2025; Shin et al., 2025; Chen et al., 2025b; Hanson et al., 2025; Zhang et al., 2026; Ding et al., 2026; Chen et al., 2026a). In online mapping, however, refinement remains fragmented across Gaussian attribute groups despite their shared visibility structure. CGO instead restructures refinement around a shared visibility-aligned execution domain, coalescing heterogeneous attribute updates into a coordinated execution path.

3 Method

Figure 2 shows our method within an online Gaussian mapping backbone that estimates poses, maintains the current map, and refines visible Gaussians through differentiable RGB/depth rendering. GeoBirth acts at Gaussian formation, converting local metric geometry into surface-aligned orientation and reliability-adaptive anisotropic support, thereby improving the geometric state of newborn Gaussians before refinement. CGO acts during refinement, reorganizing Gaussian updates over a shared visible-Gaussian workset to reduce repeated execution and state access. Thus, GeoBirth reduces the refinement demand created during formation, while CGO reduces the execution cost of processing that demand. Sections 3.1–3.3 detail the backbone, GeoBirth, and CGO.

3.1 Online Gaussian Mapping Backbone

We use the standard anisotropic 3DGS representation (Kerbl et al., 2023). A Gaussian $G_i = (\mathbf{p}_i, \mathbf{f}_i^{\mathrm{dc}}, \mathbf{f}_i^{\mathrm{sh}}, o_i, \boldsymbol{\ell}_i, \mathbf{q}_i)$ contains its center, appearance, opacity, log scale, and rotation, with

$$\boldsymbol{\Sigma}_i = \mathbf{R}(\mathbf{q}_i)\,\mathrm{diag}(\exp(2\boldsymbol{\ell}_i))\,\mathbf{R}(\mathbf{q}_i)^\top. \tag{1}$$

At keyframe t, the sensing front end provides a posed observation, a LiDAR-derived metric support set $\mathcal{P}_t$, and newborn candidates $\mathcal{B}_t$. Let Φ denote Gaussian formation and $\mathcal{U}$ the refinement operator:

$$\mathcal{G}_t^{(0)} = \mathcal{G}_{t-1} \cup \Phi(\mathcal{B}_t; \mathcal{P}_t), \qquad \mathcal{G}_t^{(k+1)} = \mathcal{U}\Big(\mathcal{G}_t^{(k)}, \nabla\mathcal{L}_{t,k}; \mathbf{v}_{t,k}\Big), \tag{2}$$

where $\mathcal{L}_{t,k}$ is the rendering objective and $\mathbf{v}_{t,k} \in \{0,1\}^{N_{t,k}}$ is the visibility mask at refinement step k. This decomposition exposes two modular interfaces in the standard online Gaussian mapping loop: GeoBirth acts on the formation operator Φ, while CGO reorganizes the execution of refinement $\mathcal{U}$.

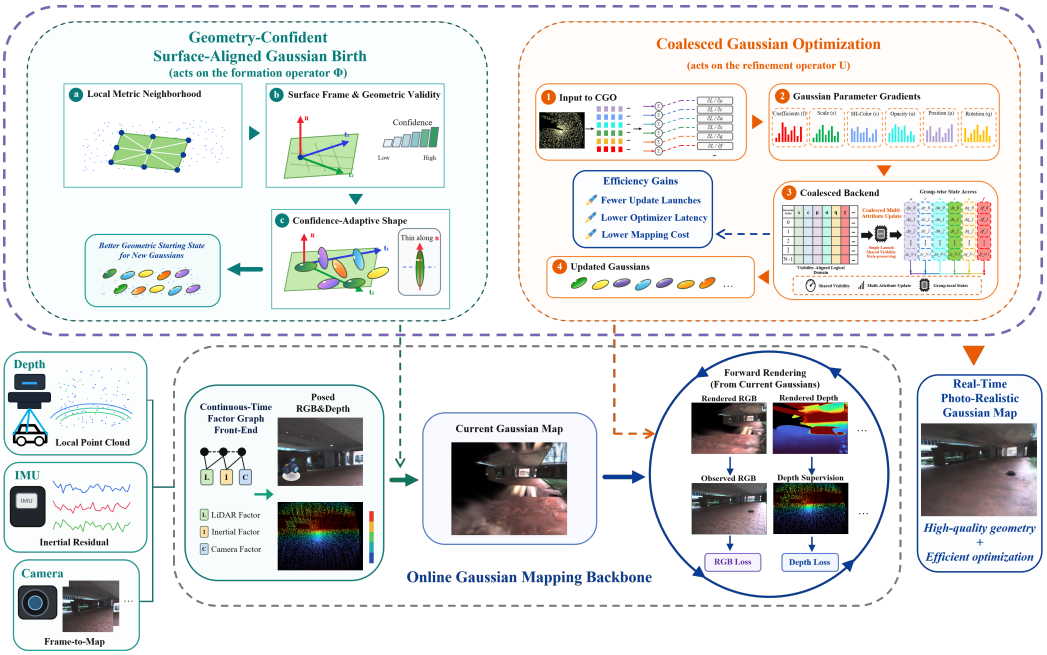

Figure 2: **Pipeline.** The front end provides posed RGB observations and metric 3D candidates. GeoBirth forms surface-aligned Gaussians with confidence-adaptive anisotropic support; differentiable rendering provides visibility and gradients, which CGO processes in one coalesced multi-attribute update.

We denote the visible-Gaussian workload by $W_{t,k} = \|\mathbf{v}_{t,k}\|_0$ and decompose online mapping cost as

$$T_{\text{map}} = T_{\text{form}} + \sum_k \left(T^{(k)}_{\text{render}} + T^{(k)}_{\text{backward}} + T^{(k)}_{\text{update}} \right) + T_{\text{other}}. \tag{3}$$

Here, $T^{(k)}_{\text{update}}$ is the update cost associated with the visible refinement workset $W_{t,k}$.

GeoBirth determines $(\mathbf{q}_i, \boldsymbol{\ell}_i)$ during Φ, aligning Gaussian covariance with local surface structure and shaping its projected support. These formation choices influence the corrections and visible refinement workload $W_{t,k}$ encountered downstream. Differentiable rendering then produces $\mathbf{v}_{t,k}$ and $\nabla \mathcal{L}_{t,k}$, which CGO consumes through $\mathcal{U}$. GeoBirth therefore acts upstream on refinement demand, while CGO targets the execution cost $T^{(k)}_{\text{update}}$ of processing that demand.

3.2 Geometry-Conditioned Surface-Aligned Gaussian Birth

GeoBirth uses the local metric points around each newborn Gaussian to estimate the orientation of the underlying surface and the confidence of this estimate. The local surface normal and tangent directions determine how the Gaussian is oriented, while the geometric confidence measures how consistently the nearby points support the same local surface. This confidence then controls the strength of anisotropic scale shaping: high-confidence geometry produces a flatter Gaussian aligned with the surface, whereas lower confidence leads to a milder shape adjustment. In this way, GeoBirth converts local metric structure directly into a surface-aligned orientation and scale for Gaussian formation.

Local geometry estimation. For candidate $\mathbf{p}_i$, let $(\mathbf{q}^0_i, \boldsymbol{\ell}^0_i)$ denote its pre-shaping orientation and log scale, with $\boldsymbol{\ell}^0_i = \ell^0_i \mathbf{1}$ and $s^0_i = \exp(\ell^0_i)$. GeoBirth retrieves up to $K = 16$ nearest metric

candidates from the current birth batch $\mathcal{B}_t$, forming $\mathcal{N}_i$ with $n_i = |\mathcal{N}_i|$, and computes

$$\bar{\mathbf{p}}_i = \frac{1}{n_i} \sum_{\mathbf{p} \in \mathcal{N}_i} \mathbf{p}, \qquad \mathbf{C}_i = \frac{1}{n_i} \sum_{\mathbf{p} \in \mathcal{N}_i} (\mathbf{p} - \bar{\mathbf{p}}_i)(\mathbf{p} - \bar{\mathbf{p}}_i)^\top. \tag{4}$$

Let $\lambda_{i,0} \le \lambda_{i,1} \le \lambda_{i,2}$ and $\mathbf{e}_{i,0}, \mathbf{e}_{i,1}, \mathbf{e}_{i,2}$ be the ordered eigenpairs, with $r_i = \max_{\mathbf{p} \in \mathcal{N}_i} \|\mathbf{p} - \mathbf{p}_i\|_2$ and $\rho_i = \lambda_{i,0}/(\lambda_{i,1} + 10^{-12}\,\mathrm{m}^2)$. The neighborhood is geometrically valid when

$$h_i = \mathbf{1}\big[n_i \ge 8,\ r_i \le 0.50\,\mathrm{m},\ \lambda_{i,1} > 10^{-12}\,\mathrm{m}^2,\ \rho_i \le 0.25\big]. \tag{5}$$

Surface-aligned orientation. For $h_i = 1$, the smallest-eigenvalue direction defines the surface normal, oriented toward the camera center $\mathbf{c}_t$:

$$\mathbf{n}_i = \begin{cases} \mathbf{e}_{i,0}, & \mathbf{e}_{i,0}^\top(\mathbf{c}_t - \mathbf{p}_i) \ge 0, \\ -\mathbf{e}_{i,0}, & \text{otherwise.} \end{cases}$$

Using $\mathbf{e}_{i,2}$ as the initial tangent, we construct

$$\begin{aligned} \tilde{\mathbf{t}}_i &= \mathrm{normalize}\big(\mathbf{e}_{i,2} - \mathbf{n}_i \mathbf{n}_i^\top \mathbf{e}_{i,2}\big), \\ \mathbf{b}_i &= \mathrm{normalize}(\mathbf{n}_i \times \tilde{\mathbf{t}}_i), \qquad \mathbf{t}_i = \mathbf{b}_i \times \mathbf{n}_i, \\ \mathbf{R}_i &= [\mathbf{t}_i, \mathbf{b}_i, \mathbf{n}_i]. \end{aligned} \tag{6}$$

The resulting orientation is $\mathbf{q}_i = \mathrm{quat}(\mathbf{R}_i)$ for $h_i = 1$, and $\mathbf{q}_i = \mathbf{q}_i^0$ otherwise.

Reliability-adaptive anisotropic support. We quantify neighborhood reliability with a geometric confidence score c_i, combining planarity, compactness, and neighborhood support:

$$\begin{aligned} q_i^p &= \mathrm{clip}\Big(1 - \frac{\rho_i}{0.25}, 0, 1\Big), & q_i^r &= \mathrm{clip}\Big(1 - \frac{r_i}{0.50\,\mathrm{m}}, 0, 1\Big), \\ q_i^s &= \mathrm{clip}\Big(\frac{n_i - 8}{8}, 0, 1\Big), & c_i &= 0.4q_i^p + 0.4q_i^r + 0.2q_i^s, \\ \alpha_i &= h_i\ \mathrm{clip}\Big(\frac{c_i - 0.85}{0.10}, 0, 1\Big). \end{aligned} \tag{7}$$

Here, h_i determines whether surface alignment is applied, whereas α_i controls the strength of anisotropic shaping.

Let $\tilde{d}_i$ be the median positive candidate-to-neighbor distance. The tangent extent, normal ratio, and final log scale are

$$\begin{aligned} s_i^{\parallel} &= \mathrm{clip}\Big(\max(s_i^0, \tilde{d}_i), 10^{-4}\,\mathrm{m}, 0.20\,\mathrm{m}\Big), \\ \eta_i &= 1 + (0.15 - 1)\alpha_i, \\ \boldsymbol{\ell}_i &= (1 - \alpha_i)\boldsymbol{\ell}_i^0 + \alpha_i \log \begin{bmatrix} s_i^{\parallel} \\ s_i^{\parallel} \\ \mathrm{clip}\Big(\eta_i s_i^{\parallel}, 10^{-4}\,\mathrm{m}, s_i^{\parallel}\Big) \end{bmatrix}. \end{aligned} \tag{8}$$

Reliable surface evidence therefore produces stronger tangent–normal anisotropy, while lower reliability yields progressively milder shaping. At maximum confidence, the target normal-to-tangent ratio reaches 0.15.

Geometric effect. GeoBirth provides newborn Gaussians with a geometry that is already consistent with the local surface at formation. This reduces the rotation and scale adjustment required during subsequent mapping and provides a more stable geometric basis for rendering.

3.3 Coalesced Gaussian Optimization

CGO targets fragmented execution during Gaussian refinement. Although position, appearance, opacity, scale, and rotation share the same visible-Gaussian set, their updates are typically issued through separate attribute-wise paths. CGO instead organizes these updates around a shared visibility-aligned workset and executes them through a coalesced multi-attribute path. As illustrated in Fig. 2, rendering provides the visibility and gradients consumed by this refinement stage.

Shared visible-Gaussian workset. We use the ordered attribute groups $\mathcal{A} = (\mathrm{xyz}, \mathrm{dc}, \mathrm{sh}, \mathrm{opacity}, \mathrm{scale}, \mathrm{rot})$ with widths $(3, 3, 45, 1, 3, 4)$, giving $D = 59$ scalars per Gaussian. Each group maintains its parameter, gradient, first- and second-moment states, and learning rate $\gamma_{a,k}$, while all groups share $\mathbf{v}_{t,k}$. CGO organizes these states around one visible-Gaussian workset, avoiding repeated traversal and scheduling of the same active rows.

Visibility-aligned coalescing. Each group descriptor stores its parameter, gradient, and moment-state addresses together with width d_a, logical offset, and learning rate. For ordered group a, let $O_a = N_{t,k} \sum_{b<a} d_b$. At each refinement step, CGO traverses the $DN_{t,k} = 59N_{t,k}$ logical entries in one coalesced dispatch. For $O_a \le x < O_a + d_a N_{t,k}$,

$$r = x - O_a, \qquad i = \left\lfloor \frac{r}{d_a} \right\rfloor, \qquad j = r \bmod d_a, \tag{9}$$

which identifies Gaussian row i and component j. Visibility $\mathbf{v}_{t,k}[i]$ is tested before moment-state access, so scheduling and workset traversal are shared across attributes while their optimization states remain group-specific.

Group-local update. For visible row i, component j of group a, and gradient g_{aij}, CGO applies

$$\begin{aligned} m^+_{aij} &= \beta_1 m_{aij} + (1-\beta_1) g_{aij}, \\ u^+_{aij} &= \beta_2 u_{aij} + (1-\beta_2) g^2_{aij}, \\ \theta^+_{aij} &= \theta_{aij} - \gamma_{a,k} \frac{m^+_{aij}}{\sqrt{u^+_{aij}} + \epsilon}, \end{aligned} \qquad \beta_1 = 0.9,\ \beta_2 = 0.999,\ \epsilon = 10^{-15}. \tag{10}$$

Visible rows enter the update. By coalescing attribute-wise update paths into a single dispatch, CGO reduces refinement execution overhead within $T^{(k)}_{\mathrm{update}}$.

Optimization effect. The shared workset removes repeated traversal and scheduling of the same active Gaussians across attribute groups. Indexing, state access, and update computation are therefore concentrated into a coordinated execution path, reducing the update cost $T^{(k)}_{\mathrm{update}}$.

4 Experiments

4.1 Experimental Setup

Datasets and Metrics. Our main evaluation uses 12 indoor and outdoor sequences from M2DGR (Yin et al., 2022), R3LIVE (Lin & Zhang, 2022), and FAST-LIVO2 (Zheng et al., 2025). Cross-backbone transfer experiments additionally use KITTI (Geiger et al., 2012) and TUM RGB-D (Sturm et al., 2012). We report PSNR (P)↑, SSIM (S)↑, LPIPS (L)↓, Mapping Time↓, and Sustained System Throughput↑. Dataset-level results use unweighted macro-averages over sequences. Unless stated otherwise, the best, second-best, and third-best results are highlighted in red, orange, and yellow, respectively.

Baselines and Evaluation Protocol. We compare rendering quality with MM3DGS-SLAM, VarSplat, SGAD-SLAM, Gaussian-LIC, and Gaussian-LIC2 (Sun et al., 2024; Tran & Kosecka, 2026; Hu & Han, 2026; Lang et al., 2025; 2026). A matched 2×2 study on Gaussian-LIC2 isolates GeoBirth at formation and CGO at refinement. For cross-backbone evaluation, we further transfer both modules to MonoGS, VarSplat, SGAD-SLAM, PVG, and GS-LIVM, spanning monocular, RGB-D, RGB–LiDAR, and LiDAR–RGB–IMU settings (Matsuki et al., 2024; Tran & Kosecka, 2026; Hu & Han, 2026; Chen et al., 2026b; Xie et al., 2025). All paired variants share observations, poses, evaluation views, and iteration budgets.

Implementation Details. All experiments run on an Intel Core i9 CPU and a single NVIDIA RTX 4090 GPU. GeoBirth computes geometric confidence from local planarity, spatial compactness, and neighborhood support with weights $0.4/0.4/0.2$.

Table 1: **Matched evaluation across Gaussian mapping backbones and sensing configurations.**

Input	Backbone / Eval. Scope	Rendering Quality									Online Efficiency		
		Base			Ours+Base			Δ			Base→Ours+Base		
		P↑	S↑	L↓	P↑	S↑	L↓	ΔP	ΔS	ΔL	Opt. GPU (ms)	Launches	Mapping (s)
LiDAR+RGB +IMU	**Gaussian-LIC2** 12 seq.	20.692	.7160	.2390	**21.041**	**.7216**	.2501	**+.349**	**+.0056**	+.0111	.3720→**.2298** (−38.23%)	6→**1**	135.41→**119.50** (−11.75%)
	GS-LIVM FAST-LIVO2 ×3	6.175	.3735	.7186	**6.408**	**.3940**	**.7135**	**+.233**	**+.0205**	**-.0051**	.0845→**.0063** (−92.5%)	40→**1**	12.61→**12.32** (−2.30%)
RGB+LiDAR	**PVG** KITTI-MOT 0006	24.280	.9291	.0743	**24.434**	**.9301**	**.0733**	**+.154**	**+.0010**	**-.0010**	.2642→**.0707** (−73.2%)	63→**1**	132.27→**117.64** (−11.06%)
RGB-D	**SGAD-SLAM** TUM ×3	12.378	.4921	.3914	**12.483**	**.4933**	**.3849**	**+.105**	**+.0012**	**-.0065**	.1786→**.0346** (−80.6%)	35→**1**	11.53→**11.30** (−1.99%)
	VarSplat TUM ×3	26.508	.9078	.0901	**26.611**	**.9081**	**.0896**	**+.103**	**+.0003**	**-.0005**	.2070→**.0397** (−80.8%)	42→**1**	206.72→**205.44** (−0.62%)
RGB-D	**MonoGS*** 6 seq.	7.980	.0517	.7606	**9.613**	**.1802**	**.7137**	**+1.633**	**+1.285**	**-.0469**	.1888→**.0350** (−81.5%)	35→**1**	12.85→**6.74** (−47.55%)

Sequences. Gaussian-LIC2: M2DGR, R3LIVE, and FAST-LIVO2 (12 seq.); GS-LIVM: FAST-LIVO2 `Bright_Screen_Wall`, `CBD_Building_01`, `CBD_Building_03`; PVG: KITTI-MOT `0006`; SGAD-SLAM/VarSplat: TUM RGB-D `fr1/desk`, `fr2/xyz`, `fr3/long_office_household`; MonoGS: M2DGR `room03`; R3LIVE `degenerate_seq_01`, `hku_campus_seq_01`, `hku_park_00`, `hku_park_01`; FAST-LIVO2 `CBD_Building_03`.

*MonoGS natively takes RGB input; in our matched evaluation, we augment RGB with one depth channel directly provided by the dataset.

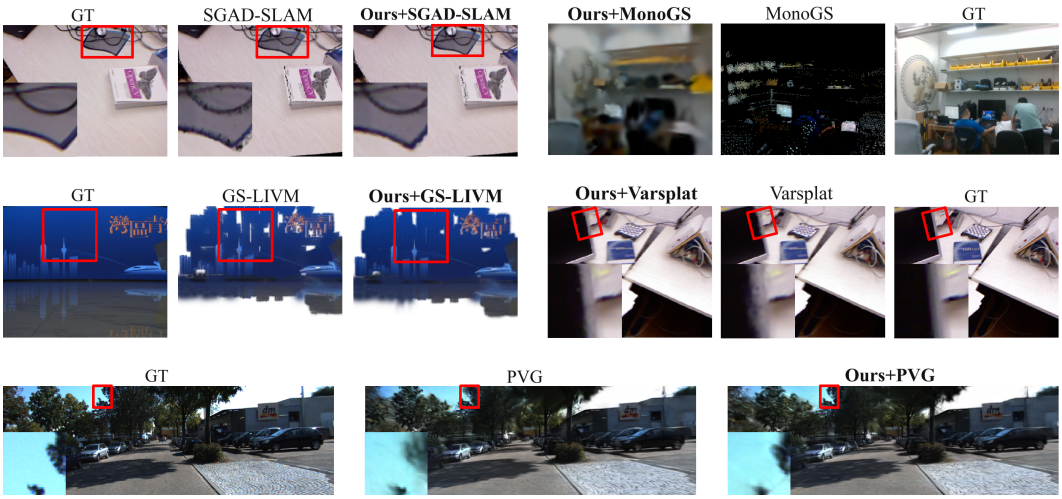

Figure 3: **Qualitative transfer across representative Gaussian backbones.**

4.2 Cross-Backbone Transferability

Quantitative transfer. Table 1 shows that FormFlow-GS transfers across substantially different sensing configurations and mapping architectures rather than depending on a particular backbone. Within this evaluation, Ours+Gaussian-LIC2 achieves the strongest overall quality–efficiency trade-off. Across all evaluated backbones, optimizer GPU time decreases and the multi-attribute updates are reduced to a single launch. Consistent optimizer-level gains indicate transfer across heterogeneous execution pipelines, while variation in end-to-end speedup reflects backbone-specific costs outside refinement. Together, these results support the modularity of the proposed formation–refinement design across Gaussian mapping systems.

Visual transfer analysis. Figure 3 complements Table 1 with qualitative results from the same backbone–dataset configurations. Across VarSplat, SGAD-SLAM, PVG, and GS-LIVM, GeoBirth preserves finer local details and sharper boundaries. For MonoGS, both paired variants use the same dataset-provided depth channel, while the variant with our modules achieves higher rendering quality. These consistent visual improvements further support cross-backbone transferability across different sensing and mapping pipelines.

4.3 FORMATION–REFINEMENT INTERACTION

We isolate the effects of Gaussian formation and refinement execution through a matched 2×2 study over all 12 sequences. GeoBirth and CGO are varied independently, revealing how formation quality affects downstream refinement and how CGO reduces the cost of executing it.

Controlled effects. Table 2 isolates the roles of formation and refinement on Gaussian-LIC2. GeoBirth retains its PSNR/SSIM benefit under either refinement path, whereas CGO lowers Mapping Time under either formation strategy, indicating that their gains originate from distinct stages of the pipeline. Importantly, CGO preserves the formation-side benefit of GeoBirth: E11 combines the strongest PSNR/SSIM with the lowest Mapping Time, supporting a complementary formation–refinement design.

Table 2: **Controlled 2×2 decomposition.** GeoBirth varies formation and CGO varies refinement execution; all other settings are matched.

Variant	Formation	Refinement	PSNR↑	SSIM↑	LPIPS↓	Mapping (s)↓	Sustained FPS↑
E00	Native	Native	20.692	.7160	**.2390**	135.41	1034.16
E10	GeoBirth	Native	20.847	.7205	.2550	131.50	**1108.41**
E01	Native	CGO	20.675	.7184	.2550	124.72	1057.77
E11	GeoBirth	CGO	**21.041**	**.7216**	.2501	**119.50**	1108.22

Formation effect. Figure 4 illustrates the direct geometric effect of GeoBirth at Gaussian formation. Native newborns are close to isotropic, whereas GeoBirth uses the local metric surface to orient the Gaussian and expand its support along the tangent plane. The resulting Gaussians better follow local surface structure. The corresponding image crops show sharper boundaries and finer details; under challenging illumination, the reconstruction also remains closer to the ground truth when RGB appearance cues are less reliable.



Figure 4: **Local formation and rendering effects of GeoBirth.** Representative examples from FAST-LIVO2 (Zheng et al., 2025) and R3LIVE (Lin & Zhang, 2022).

Table 3: **Refinement demand.**

Metric	w/o CGO E10 vs. E00	w/ CGO E11 vs. E01
Geometric correction		
Rotation-update magnitude	8.95% ↓	8.59% ↓
Scale-update magnitude	7.76% ↓	7.19% ↓
Newborn rotation correction	4.95% ↓	3.59% ↓
Newborn scale correction	10.38% ↓	8.75% ↓
Refinement workload		
Mean visible-Gaussian workload	*N/A*	11.35% ↓
Per-visible-Gaussian optimizer cost	*N/A*	0.19% ↓

Table 4: **Final-map compactness.**

Sequence	Base	+GeoBirth	Δ
Bright Screen Wall	0.408	**0.311**	−23.75%
M2DGR room03	1.878	**1.673**	−10.94%
M2DGR room02	1.385	**1.254**	−9.46%
M2DGR room01	1.544	**1.414**	−8.44%
CBD Building 03	2.011	**1.847**	−8.16%
R3LIVE campus00	1.338	**1.272**	−4.88%
R3LIVE park00	1.489	**1.442**	−3.20%
12-seq. pooled	15.121	**14.324**	−5.27%

Counts are in millions; Base denotes Gaussian-LIC2, Pooled is the total over 12 sequences.

Refinement demand. Table 3 isolates the formation effect through paired comparisons under the same refinement backend (E10 vs. E00 and E11 vs. E01). For each pair, we accumulate the rotation and scale updates applied over the same mapping interval, and repeat the measurement using

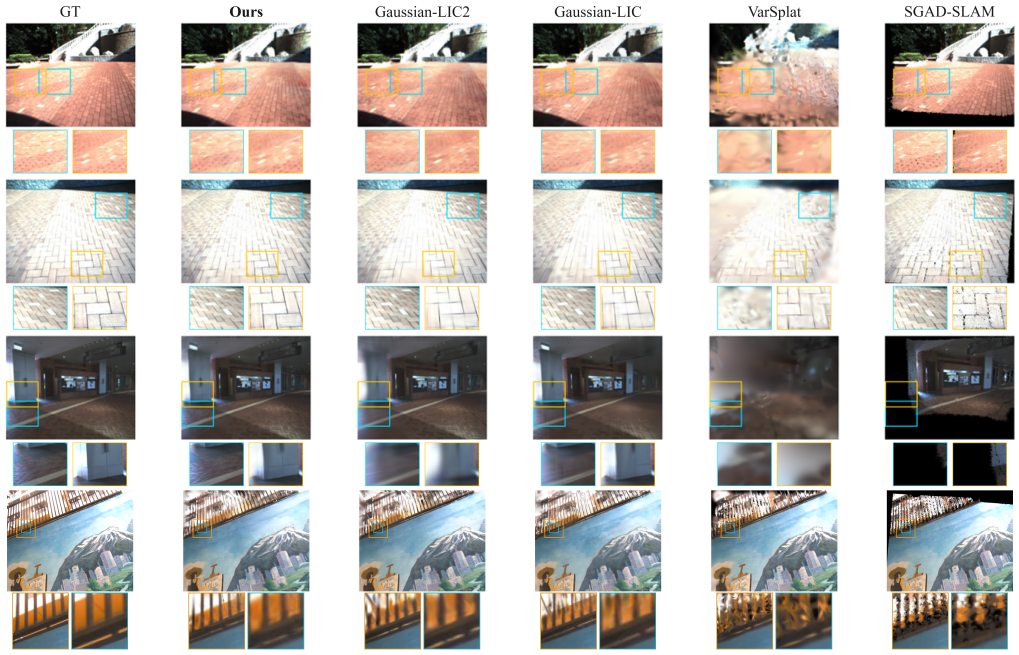

Figure 5: **Qualitative rendering under challenging illumination** on FAST-LIVO2 (Zheng et al., 2025) and R3LIVE (Lin & Zhang, 2022)

newly inserted Gaussians. GeoBirth consistently requires less geometric correction, showing that its surface-aligned initialization provides a better geometric starting state and reduces the rotation and scale adjustment needed during subsequent mapping. This also explains the Mapping-Time reduction in Table 2: with less correction required after formation, GeoBirth reduces the refinement demand even before the backend execution is changed.

Workload effect. The reduced correction demand is also reflected in a smaller visible refinement workset. With CGO fixed, GeoBirth lowers the mean $W_{t,k} = \|\mathbf{v}_{t,k}\|_0$, so fewer visible Gaussians need to be processed at each refinement step. This reduces the amount of Gaussian state presented to the backend and explains the additional Mapping-Time benefit of GeoBirth. CGO then operates on this smaller workset by coalescing the remaining multi-attribute updates, further reducing update overhead. This division follows Eq. 3: GeoBirth reduces the workload associated with $T^{(k)}_{\text{update}}$, while CGO reduces the cost of executing the remaining updates.

4.4 Rendering under Challenging Illumination

Full per-sequence rendering and mapping-efficiency results are reported in Appendix Tables 8 and 9, while Figure 5 focuses on challenging illumination cases. Across representative outdoor scenes, FormFlow-GS preserves clearer local textures and structural boundaries when RGB appearance becomes less reliable, consistent with GeoBirth establishing surface-aligned newborn Gaussians from reliable metric geometry.

5 Conclusion

Our results reveal a broader principle for online Gaussian mapping: efficiency is governed by both the geometric state entering refinement and the cost of processing it. **FormFlow-GS** address these two aspects complementarily, with gains that persist across multimodal benchmarks and diverse mapping backbones. Controlled studies further show that downstream correction and update overhead can be reduced together. Future work will explore resource-adaptive mapping for dynamic scenes and compute-constrained robotic platforms.

AI USE STATEMENT

Large language models (LLMs) were used solely for language editing and manuscript formatting. Specifically, they assisted with grammar correction, stylistic refinement, and LaTeX formatting adjustments. LLMs were not used for research ideation, methodology design, experiment design or execution, data analysis, or the generation or interpretation of results. All AI-assisted edits were reviewed by the authors, who take full responsibility for the content of this paper.

ETHICS STATEMENT

This work uses publicly available research benchmarks and does not involve the collection of new human-subject or personally identifiable data. The datasets are used solely for research evaluation of online Gaussian mapping, and no new dataset or user-facing decision system is introduced. We follow the applicable dataset licenses and the ICLR Code of Ethics.

REPRODUCIBILITY STATEMENT

We provide the algorithmic formulation, implementation details, and key hyperparameters of GeoBirth and CGO in the main paper and appendix. All quantitative comparisons follow matched evaluation protocols on publicly available benchmarks, with controlled formation–refinement studies used to isolate the contribution of each component. Additional experimental settings, diagnostic analyses, and qualitative results are documented in the appendix to facilitate reproduction of the reported results. Code is available at `https://anonymous.4open.science/r/formflow-gs-review/`.

REFERENCES

Danpeng Chen, Hai Li, Weicai Ye, Yifan Wang, Weijian Xie, Shangjin Zhai, Nan Wang, Haomin Liu, Hujun Bao, and Guofeng Zhang. PGSR: Planar-based gaussian splatting for efficient and high-fidelity surface reconstruction. *IEEE Transactions on Visualization and Computer Graphics*, 31(9):6100–6111, 2025a.

Jingxiang Chen, Mohamed Ibrahim, and Yang Liu. VkSplat: High-performance 3DGS training in Vulkan compute. In *Eurographics 2026 - Short Papers*. The Eurographics Association, 2026a. doi: 10.2312/egs.20261024.

Youyu Chen, Junjun Jiang, Kui Jiang, Xiao Tang, Zhihao Li, Xianming Liu, and Yinyu Nie. DashGaussian: Optimizing 3D gaussian splatting in 200 seconds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11146–11155, 2025b.

Yurui Chen, Chun Gu, Junzhe Jiang, Xiatian Zhu, and Li Zhang. Periodic vibration gaussian: Dynamic urban scene reconstruction and real-time rendering. *International Journal of Computer Vision*, 134(3):83, 2026b. doi: 10.1007/s11263-026-02740-3.

Renjie Ding, Yaonan Wang, Min Liu, Jialin Zhu, Jiazheng Wang, Jiahao Zhao, Wenting Shen, Feixiang He, and Xiang Chen. A step to decouple optimization in 3DGS. In *The Fourteenth International Conference on Learning Representations*, 2026.

Zhiwen Fan, Kevin Wang, Kairun Wen, Zehao Zhu, Dejia Xu, and Zhangyang Wang. LightGaussian: Unbounded 3D gaussian compression with 15x reduction and 200+ FPS. In *Advances in Neural Information Processing Systems*, volume 37, pp. 140138–140158, 2024.

Guangchi Fang and Bing Wang. Mini-splatting: Representing scenes with a constrained number of gaussians. In *European Conference on Computer Vision*, pp. 165–181, 2024.

Ruilan Gao, Letian Jin, and Yu Zhang. GeoGS-SLAM: Online monocular reconstruction using gaussian splatting with geometric priors. In *2026 IEEE International Conference on Robotics and Automation*, 2026.

Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the KITTI vision benchmark suite. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3354–3361, 2012. doi: 10.1109/CVPR.2012.6248074.

Sharath Girish, Kamal Gupta, and Abhinav Shrivastava. EAGLES: Efficient accelerated 3D gaussians with lightweight EncodingS. In *European Conference on Computer Vision*, pp. 54–71, 2024.

Antoine Guédon and Vincent Lepetit. SuGaR: Surface-aligned gaussian splatting for efficient 3D mesh reconstruction and high-quality mesh rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5354–5363, 2024.

Antoine Guédon, Tomoki Ichikawa, Kohei Yamashita, and Ko Nishino. MAtCha gaussians: Atlas of charts for high-quality geometry and photorealism from sparse views. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6001–6011, 2025.

Alex Hanson, Allen Tu, Geng Lin, Vasu Singla, Matthias Zwicker, and Tom Goldstein. Speedy-Splat: Fast 3D gaussian splatting with sparse pixels and sparse primitives. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21537–21546, 2025.

Qiqi Hou, Randall Rauwendaal, Zifeng Li, Hoang Le, Farzad Farhadzadeh, Fatih Porikli, Alexei Bourd, and Amir Said. Sort-free gaussian splatting via weighted sum rendering. In *The Thirteenth International Conference on Learning Representations*, 2025.

Pengchong Hu and Zhizhong Han. SGAD-SLAM: Splatting gaussians at adjusted depth for better radiance fields in RGBD SLAM. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18934–18945, 2026.

Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2D gaussian splatting for geometrically accurate radiance fields. In *ACM SIGGRAPH 2024 Conference Papers*, 2024. doi: 10.1145/3641519.3657428.

Kaiwen Jiang, Venkataram Sivaram, Cheng Peng, and Ravi Ramamoorthi. Geometry field splatting with gaussian surfels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5752–5762, 2025.

Nikhil Keetha, Jay Karhade, Krishna Murthy Jatavallabhula, Gengshan Yang, Sebastian Scherer, Deva Ramanan, and Jonathon Luiten. SplaTAM: Splat, track & map 3D gaussians for dense RGB-D SLAM. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21357–21366, 2024.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3D gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4):1–14, 2023. doi: 10.1145/3592433.

Xiaolei Lang, Laijian Li, Chenming Wu, Chen Zhao, Lina Liu, Yong Liu, Jiajun Lv, and Xingxing Zuo. Gaussian-LIC: Real-time photo-realistic SLAM with gaussian splatting and LiDAR-inertial-camera fusion. In *2025 IEEE International Conference on Robotics and Automation*, pp. 8500–8507, 2025.

Xiaolei Lang, Jiajun Lv, Kai Tang, Laijian Li, Jianxin Huang, Lina Liu, Yong Liu, and Xingxing Zuo. Gaussian-LIC2: LiDAR-inertial-camera gaussian splatting SLAM. *The International Journal of Robotics Research*, 2026.

Dongen Li, Yi Liu, Junqi Liu, Zewen Sun, Zefan Huang, Shuo Sun, Jiahui Liu, Chengran Yuan, Hongliang Guo, Francis E. H. Tay, and Marcelo H. Ang. RMGS-SLAM: Real-time multi-sensor gaussian splatting SLAM. *arXiv preprint arXiv:2604.12942*, 2026.

Mingrui Li, Shuhong Liu, Heng Zhou, Guohao Zhu, Na Cheng, Tianchen Deng, and Hongyu Wang. SGS-SLAM: Semantic gaussian splatting for neural dense SLAM. In *European Conference on Computer Vision*, pp. 163–179, 2024a.

Yanyan Li, Chenyu Lyu, Yan Di, Guangyao Zhai, Gim Hee Lee, and Federico Tombari. GeoGaussian: Geometry-aware gaussian splatting for scene rendering. In *European Conference on Computer Vision*, pp. 441–457, 2024b.

Jiarong Lin and Fu Zhang. R3LIVE: A robust, real-time, RGB-colored, LiDAR-inertial-visual tightly-coupled state estimation and mapping package. In *2022 International Conference on Robotics and Automation*, pp. 10672–10678, 2022.

Tao Lu, Mulin Yu, Linning Xu, Yuanbo Xiangli, Limin Wang, Dahua Lin, and Bo Dai. Scaffold-GS: Structured 3D gaussians for view-adaptive rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20654–20664, 2024.

Saswat Subhajyoti Mallick, Rahul Goel, Bernhard Kerbl, Markus Steinberger, Francisco Vicente Carrasco, and Fernando De La Torre. Taming 3DGS: High-quality radiance fields with limited resources. In *SIGGRAPH Asia 2024 Conference Papers*. Association for Computing Machinery, 2024. doi: 10.1145/3680528.3687694.

Hidenobu Matsuki, Riku Murai, Paul H. J. Kelly, and Andrew J. Davison. Gaussian splatting SLAM. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18039–18048, 2024.

Hidenobu Matsuki, Gwangbin Bae, and Andrew J. Davison. 4DTAM: Non-rigid tracking and mapping via dynamic surface gaussians. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 26921–26932, 2025.

K. L. Navaneet, Kossar Pourahmadi Meibodi, Soroush Abbasi Koohpayegani, and Hamed Pirsiavash. CompGS: Smaller and faster gaussian splatting with vector quantization. In *European Conference on Computer Vision*, pp. 330–349, 2024.

Simon Niedermayr, Josef Stumpfegger, and Rüdiger Westermann. Compressed 3D gaussian splatting for accelerated novel view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10349–10358, 2024.

Cong Ruan, Yuesong Wang, Tao Guan, Bin Zhang, and Lili Ju. IndoorGS: Geometric cues guided gaussian splatting for indoor scene reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 844–853, 2025.

Seungjoo Shin, Jaesik Park, and Sunghyun Cho. Locality-aware gaussian compression for fast and high-quality rendering. In *The Thirteenth International Conference on Learning Representations*, 2025.

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.

Lisong C. Sun, Neel P. Bhatt, Jonathan C. Liu, Zhiwen Fan, Zhangyang Wang, Todd E. Humphreys, and Ufuk Topcu. MM3DGS SLAM: Multi-modal 3D gaussian splatting for SLAM using vision, depth, and inertial measurements. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 10159–10166, 2024.

Anh Thuan Tran and Jana Kosecka. VarSplat: Uncertainty-aware 3D gaussian splatting for robust RGB-D SLAM. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 26072–26082, 2026.

Matias Turkulainen, Xuqian Ren, Iaroslav Melekhov, Otto Seiskari, Esa Rahtu, and Juho Kannala. DN-Splatter: Depth and normal priors for gaussian splatting and meshing. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 2421–2431, 2025.

Mirko Usuelli, David Rapado-Rincon, Gert Kootstra, and Matteo Matteucci. AgriGS-SLAM: Orchard mapping across seasons via multi-view gaussian splatting SLAM. *IEEE Robotics and Automation Letters*, 11(6):7102–7109, 2026. doi: 10.1109/LRA.2026.3685453.

Jiang Wu, Rui Li, Yu Zhu, Rong Guo, Jinqiu Sun, and Yanning Zhang. Sparse2DGS: Geometry-prioritized gaussian splatting for surface reconstruction from sparse views. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11307–11316, 2025.

Yusen Xie, Zhenmin Huang, Jin Wu, and Jun Ma. GS-LIVM: Real-time photo-realistic LiDAR-inertial-visual mapping with gaussian splatting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 26869–26878, 2025.

Haofei Xu, Songyou Peng, Fangjinhua Wang, Hermann Blum, Daniel Barath, Andreas Geiger, and Marc Pollefeys. DepthSplat: Connecting gaussian splatting and depth. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16453–16463, 2025.

Chi Yan, Delin Qu, Dan Xu, Bin Zhao, Zhigang Wang, Dong Wang, and Xuelong Li. GS-SLAM: Dense visual SLAM with 3D gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19595–19604, 2024.

Vickie Ye, Ruilong Li, Justin Kerr, Matias Turkulainen, Brent Yi, Zhuoyang Pan, Otto Seiskari, Jianbo Ye, Jeffrey Hu, Matthew Tancik, and Angjoo Kanazawa. gsplat: An open-source library for gaussian splatting. *Journal of Machine Learning Research*, 26(34):1–17, 2025.

Jie Yin, Ang Li, Tao Li, Wenxian Yu, and Danping Zou. M2DGR: A multi-sensor and multi-scenario SLAM dataset for ground robots. *IEEE Robotics and Automation Letters*, 7(2):2266–2273, 2022. doi: 10.1109/LRA.2021.3138527.

Vladimir Yugay, Yue Li, Theo Gevers, and Martin R. Oswald. Gaussian-SLAM: Photo-realistic dense SLAM with gaussian splatting. *arXiv preprint arXiv:2312.10070*, 2023.

Zicheng Zhang, Ke Wu, Xiangting Meng, Keyu Liu, Jieru Zhao, and Wenchao Ding. Flash-Mono: Feed-forward accelerated gaussian splatting monocular SLAM. In *The Fourteenth International Conference on Learning Representations*, 2026.

Hexu Zhao, Haoyang Weng, Daohan Lu, Ang Li, Jinyang Li, Aurojit Panda, and Saining Xie. On scaling up 3D gaussian splatting training. In *The Thirteenth International Conference on Learning Representations*, 2025.

Chunran Zheng, Wei Xu, Zuhao Zou, Tong Hua, Chongjian Yuan, Dongjiao He, Bingyang Zhou, Zheng Liu, Jiarong Lin, Fangcheng Zhu, Yunfan Ren, Rong Wang, Fanle Meng, and Fu Zhang. FAST-LIVO2: Fast, direct LiDAR-inertial-visual odometry. *IEEE Transactions on Robotics*, 41: 326–346, 2025.

Zheng Zhou, Yu-Jie Xiong, Jia-Chen Zhang, Chun-Ming Xia, Xihe Qiu, and Hongjian Zhan. Gradient-direction-aware density control for 3D gaussian splatting. In *The Fourteenth International Conference on Learning Representations*, 2026.

Liyuan Zhu, Yue Li, Erik Sandström, Shengyu Huang, Konrad Schindler, and Iro Armeni. Loop-Splat: Loop closure by registering 3D gaussian splats. In *2025 International Conference on 3D Vision*, pp. 156–167, 2025.

APPENDIX

A GEOBIRTH: FROM FORMATION CUE TO PERSISTENT SURFACE EFFECT

The main paper introduces GeoBirth as a reliability-aware Gaussian formation mechanism: local metric geometry provides a surface-aligned formation cue, while neighborhood reliability controls the strength of geometric shaping. We examine this design through four questions: **(i)** why use metric geometry as a formation cue, **(ii)** how does neighborhood reliability control GeoBirth, **(iii)** what geometric structure does GeoBirth establish at birth, and **(iv)** how is this birth-time geometric effect reflected in the final surface? Together, these analyses trace GeoBirth from *formation cue* and *reliability control* to *birth-time geometry* and *final-surface evidence*.

A.1 WHY USE METRIC GEOMETRY AS A FORMATION CUE?

Question and protocol. GeoBirth relies on local metric structure at insertion time, so we first ask whether that structure provides a more stable formation cue than photometric appearance under challenging illumination. We evaluate three representative sequences: *Bright Screen Wall*, *Retail Street*, and *HKU Landmark*. For photometric perturbations, only the RGB observations are modified, while all geometry metrics are computed against the same clean LiDAR depth reference. We consider moderate underexposure (gain 0.65, gamma 1.15), moderate overexposure (gain 1.30, gamma 0.93 with clipping), and a LiDAR perturbation with $1\,\mathrm{cm}$ Gaussian range noise and 5% deterministic point dropout.

Table 5: **Geometry–appearance sensitivity under controlled perturbations.** Geometry is evaluated against clean LiDAR depth and appearance against clean RGB.

	Geometry ↓		Appearance		
Condition	**Depth L1 (m)** clean: 0.2587	**RMSE (m)** clean: 0.8669	**PSNR ↑** clean: 24.40	**SSIM ↑** clean: 0.7711	**LPIPS ↓** clean: 0.1907
Underexposure	0.2445 (-5.49%)	0.8428 (-2.78%)	15.11 (-9.29 dB)	0.6413 (-0.1298)	0.2516 (+0.0609)
Overexposure	0.2647 (+2.32%)	0.8779 (+1.27%)	16.51 (-7.89 dB)	0.7141 (-0.0570)	0.2005 (+0.0098)
LiDAR noise	0.2569 (-0.70%)	0.8767 (+1.13%)	24.32 (-0.08 dB)	0.7703 (-0.0008)	0.1914 (+0.0007)

Evidence. Table 5 shows a pronounced sensitivity gap. Under under- and overexposure, PSNR decreases by 7.89–9.29 dB, with SSIM and LPIPS changing in the same unfavorable direction, whereas depth RMSE changes by at most 2.8% and Depth L1 by at most 5.5%. Conversely, the tested LiDAR corruption changes macro Depth L1 and RMSE by only -0.70% and $+1.13\%$, respectively, while appearance remains essentially unchanged.

Conclusion. Within this diagnostic range, metric geometry exhibits markedly lower sensitivity than appearance under illumination changes. This supports using local metric structure to determine newborn orientation and support, but does not assume that every local neighborhood is equally reliable. The next analysis therefore examines how GeoBirth regulates the strength of this geometric cue.

A.2 HOW DOES NEIGHBORHOOD RELIABILITY CONTROL GEOBIRTH?

Question. The key design is not merely to make newborn Gaussians anisotropic, but to control *when* and *how strongly* geometry should shape them. Figure 6 visualizes this control mechanism from activation to directional support.

Mechanism. At low confidence, scale shaping remains inactive (Fig. 6a). Once the reliability threshold is crossed, both the shaping coefficient α and the induced log-scale displacement increase with confidence (Fig. 6b). The resulting adaptation is directional: tangent-plane support grows much more strongly than normal support (Fig. 6c), which drives the normal-to-tangent support ratio

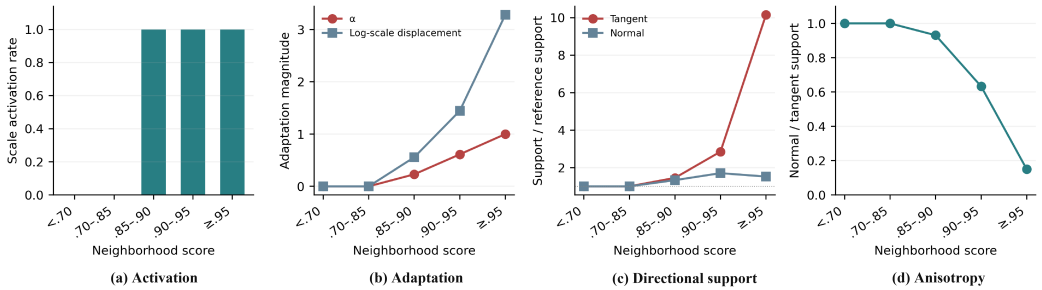

Figure 6: **Reliability-controlled formation in GeoBirth.** The four panels show, respectively, activation of scale shaping, confidence-dependent shaping strength, directional support, and the resulting normal-to-tangent support ratio.

Table 6: **Birth-time geometry of Native formation and GeoBirth.** T/N measures tangent-to-normal support, while Orientation Defined reports the fraction of GeoBirth newborns with a stable surface-normal direction.

Sequence	Native T/N	+GeoBirth T/N ↑	T/N Gain ↑	Orientation Defined ↑
Bright Screen Wall	1.0000	**1.3014**	**+30.14%**	**70.28%**
Retail Street	1.0000	**1.0513**	**+5.13%**	**17.79%**
CBD Building 03	1.0000	**1.0788**	**+7.88%**	**29.73%**

from near-isotropic values toward the full-confidence target of 0.15 (Fig. 6d). Thus, Eqs. 7 and 8 implement two distinct decisions: geometric validity determines whether a local surface frame can be used, while neighborhood reliability determines the strength and anisotropy of the subsequent scale shaping.

Conclusion. GeoBirth is therefore a reliability-modulated formation rule rather than an always-on anisotropic initializer. Having established the control mechanism, we next test whether it creates a measurable geometric change in the newborn primitives themselves.

A.3 What Geometry Does GeoBirth Establish at Birth?

Question and metrics. To directly characterize the geometric effect of GeoBirth at formation time, we measure two complementary properties of newborn Gaussians: surface-aligned anisotropy and directional structure. Let σ_{t} denote the average support within the local tangent plane and σ_{n} the support along the LiDAR-derived surface normal. We define the tangent-to-normal support ratio as

$$\mathrm{T/N} = \frac{\sigma_{\mathrm{t}}}{\sigma_{\mathrm{n}}}. \tag{11}$$

T/N close to 1 indicates approximately isotropic support, while larger values indicate stronger surface-aligned anisotropy. The *orientation-defined fraction* measures the proportion of newborn Gaussians with sufficiently distinct directional structure to establish a stable local surface-normal direction. Together, these quantities characterize surface alignment and directional structure at birth.

Evidence. Table 6 shows a measurable change in newborn Gaussian geometry. Native formation is isotropic by construction with T/N=1, whereas GeoBirth increases T/N on all three sequences, with gains from 5.13% to 30.14%. Meanwhile, 17.79%–70.28% of GeoBirth newborns acquire a stable local surface orientation. These results show that GeoBirth establishes stronger surface-aligned anisotropy together with clearer directional structure at birth.

Conclusion. GeoBirth therefore changes newborn geometry at formation, increasing surface-aligned support and well-defined local orientation. We next examine whether this birth-time geometric effect is reflected in the final explicit reconstruction.

Table 7: **Final explicit-surface reconstruction within LiDAR-observed support.** Lower is better except for F@2cm.

Sequence	Accuracy ↓		Completeness ↓		Chamfer ↓		F@2cm ↑		Normal Err. ↓	
	Native	+GeoBirth	Native	+GeoBirth	Native	+GeoBirth	Native	+GeoBirth	Native	+GeoBirth
Retail Street	22.72	**17.97**	9.92	**8.26**	16.32	**13.12**	32.77	**36.35**	30.68	**28.28**
HKU Centennial Garden 01	13.15	**12.85**	9.49	**7.90**	11.32	**10.37**	40.11	**40.72**	25.98	**24.91**
R3LIVE degenerate seq 01	24.64	**23.17**	5.98	**5.84**	15.31	**14.50**	22.23	**23.79**	23.37	**22.85**

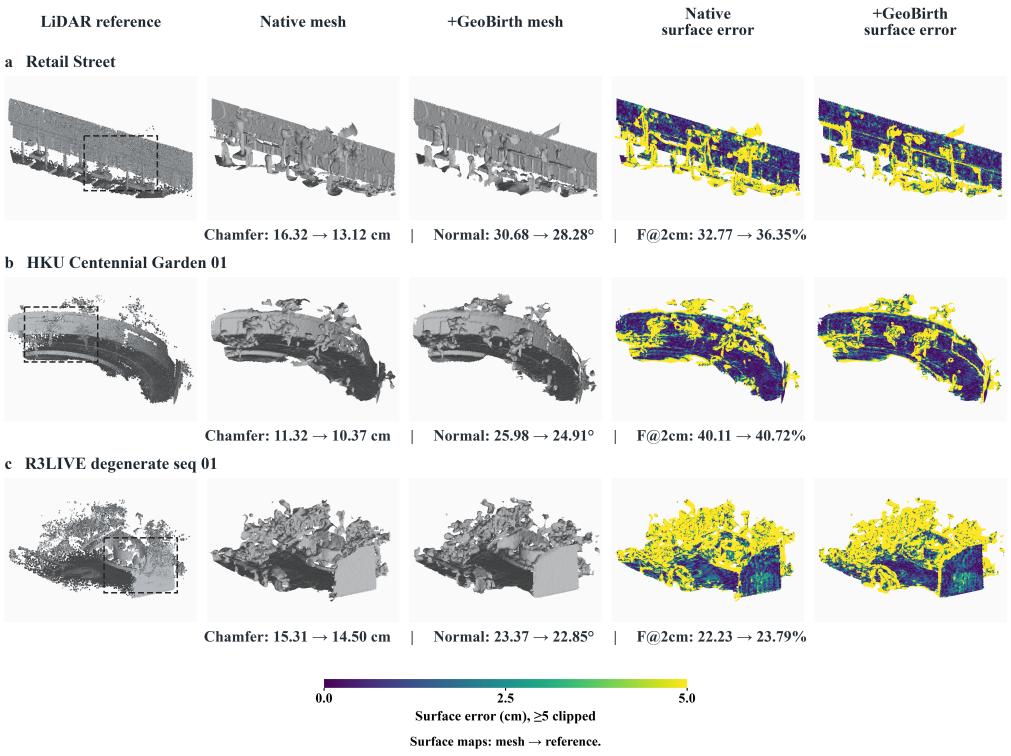

Figure 7: **Final surface reconstruction with Native formation and GeoBirth.** Mesh-to-reference error maps visualize surface discrepancy within LiDAR-observed support.

A.4 How Is the Birth-Time Effect Reflected in the Final Surface?

Question and evaluation. We finally examine whether the geometric effect introduced by GeoBirth at formation is reflected in the reconstructed surface. Within LiDAR-observed support, we evaluate mesh-to-reference Accuracy, reference-to-mesh Completeness, symmetric Chamfer distance, F@2cm, and surface-normal angular error.

Quantitative results. Table 7 shows a consistent geometric improvement across all three sequences. GeoBirth reduces Accuracy, Completeness, Chamfer distance, and normal error, while increasing F@2cm in every reported case. Thus, the formation-side geometric effect is accompanied by improved final-surface geometry.

Surface error. Figure 7 localizes the quantitative gains. GeoBirth shows fewer high-error regions and better agreement with the LiDAR reference, particularly around structural boundaries and locally complex geometry.

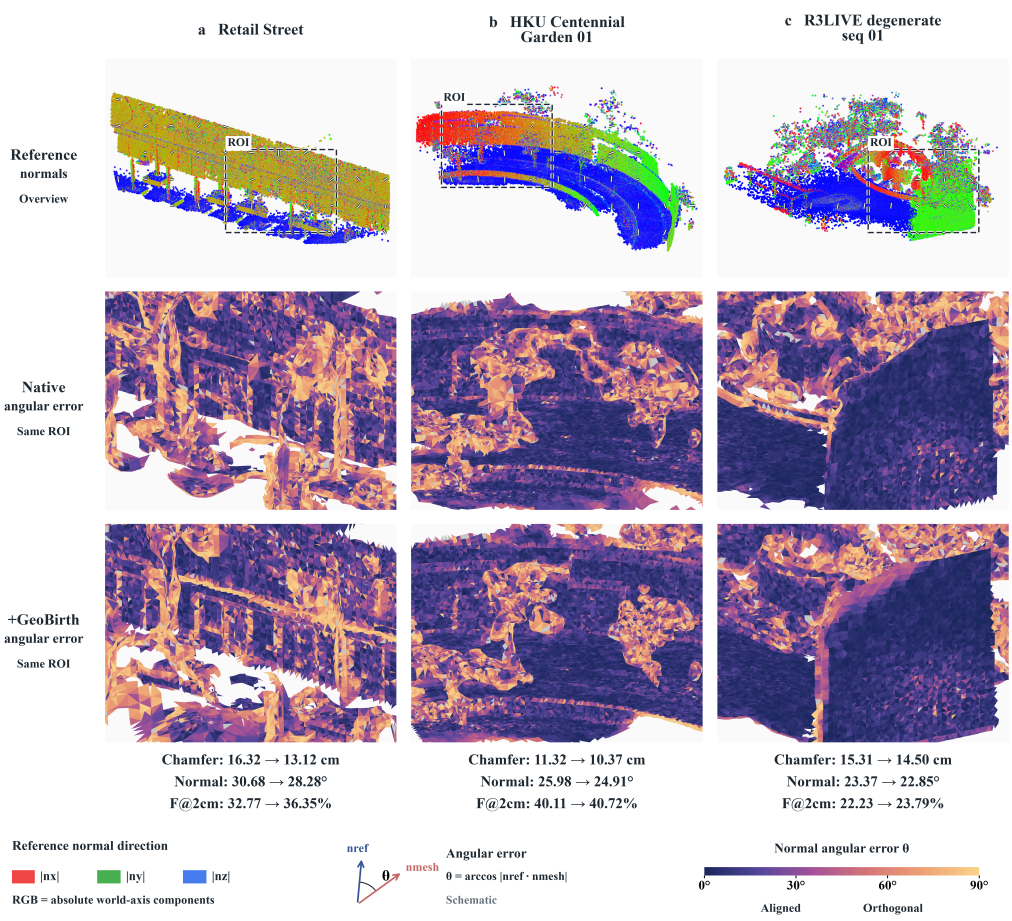

Figure 8: **Surface-normal consistency in the final reconstruction.** Angular-error maps compare Native formation and GeoBirth over matched regions using the same reference normals.

Normal consistency. Figure 8 further shows lower angular discrepancy with GeoBirth, consistent with the reduced normal error in Table 7. Together, the distance and normal results show that the surface-aligned structure established at birth is reflected in the final reconstructed geometry.

Overall conclusion. The four analyses form a progressive evidence chain: stable metric geometry motivates the formation cue; neighborhood reliability controls its strength; GeoBirth establishes measurable surface-aligned structure at birth; and the final reconstruction shows corresponding improvements in surface distance and normal consistency.

B Per-Sequence Rendering Quality and Mapping Efficiency

The main paper summarizes the aggregate and cross-backbone results. Here, we report the full per-sequence rendering quality and online mapping efficiency on the same 12 sequences from M2DGR, R3LIVE, and FAST-LIVO2. Table 8 reports PSNR, SSIM, and LPIPS across the compared methods, while Table 9 gives the corresponding Mapping Time and Sustained Throughput for Gaussian-LIC2 and Ours.

Rendering quality. Table 8 shows that FormFlow-GS consistently improves reconstruction fidelity across diverse indoor and outdoor scenes. The improvement persists across substantially different scene structures rather than being driven by a particular sequence. This supports the transferability of the proposed formation–refinement design across varied mapping scenarios.

Table 8: **In-sequence novel-view rendering quality across 12 sequences.**

Dataset	Sequence	MM3DGS-SLAM			VarSplat			SGAD-SLAM			Gaussian-LIC			Gaussian-LIC2			Ours		
		P↑	S↑	L↓	P↑	S↑	L↓	P↑	S↑	L↓	P↑	S↑	L↓	P↑	S↑	L↓	P↑	S↑	L↓
M2DGR	room01	12.270	.5540	.4580	11.980	.5177	.8059	10.550	.2888	.6701	17.040	.6970	.3650	17.450	.7210	.2820	**17.755**	**.7318**	**.2735**
M2DGR	room02	11.440	.5650	.4650	12.514	.5598	.8051	10.785	.2905	.6895	17.320	.7050	.3750	17.720	.7250	.2910	**18.081**	**.7430**	**.2763**
M2DGR	room03	12.500	.5730	.4750	11.842	.5542	.7930	9.451	.2481	.7372	17.190	.7010	.4010	17.380	.7090	**.3340**	**17.649**	**.7176**	.3438
R3LIVE	degenerate 00	19.350	.6980	.2120	14.084	.5129	.6522	12.096	.3889	.4143	21.940	.7770	.1710	21.990	.7820	**.1620**	**22.808**	**.7856**	.1755
R3LIVE	degenerate 01	18.570	.6210	.2770	14.532	.5371	.6719	12.053	.3465	.5387	22.800	.7870	.1690	22.940	**.7900**	**.1590**	**23.356**	.7872	.1767
R3LIVE	campus 00	18.700	.6210	.3190	14.707	.5344	.7663	13.142	.3432	.5504	24.380	.7870	.1620	24.340	.7870	**.1560**	**24.601**	**.7890**	.1752
R3LIVE	campus 01	16.970	.5930	.2710	12.974	.4718	.7673	13.251	.3587	.5164	20.550	.6710	.2750	20.520	.6740	**.2360**	**20.989**	**.6787**	.2605
R3LIVE	park 00	14.980	.3630	.3960	12.884	.3685	.7838	13.021	.2892	.5357	17.090	.4660	.3540	17.170	.4690	**.3400**	**17.436**	**.4815**	.3657
R3LIVE	park 01	15.790	.3920	.4090	12.459	.3679	.8200	11.806	.2237	.5718	19.450	.5290	.3400	19.460	.5290	**.3250**	**19.716**	**.5372**	.3625
FAST-LIVO2	Bright Screen Wall	23.079	.8270	.1740	18.608	.7561	.4782	15.883	.5906	.4375	24.410	.8740	.1640	24.758	.8786	.1512	**25.543**	**.8870**	**.1440**
FAST-LIVO2	CBD Building 01	22.130	.8120	.1260	12.493	.5242	.7828	14.101	.4997	.4321	25.320	.8520	.1000	25.394	.8517	**.0959**	**25.461**	**.8520**	.1020
FAST-LIVO2	CBD Building 03	13.395	.4502	.5228	13.043	.4923	.7372	11.316	.3542	.5748	18.940	.6670	.3630	**19.176**	**.6754**	**.3364**	19.102	.6690	.3460

Table 9: **Per-sequence online mapping efficiency across 12 sequences.**

Dataset	Sequence	Mapping Time (s) ↓			Sustained Throughput ↑		
		G-LIC2	Ours	Δ	G-LIC2	Ours	Δ
M2DGR	room01	39.56	**32.63**	**-17.52%**	1079.7	**1164.2**	**+7.83%**
	room02	38.41	**31.37**	**-18.33%**	1341.8	**1452.7**	**+8.27%**
	room03	101.46	**80.94**	**-20.22%**	1014.1	**1120.0**	**+10.44%**
R3LIVE	degenerate 00	38.83	**35.59**	**-8.34%**	1435.7	**1443.0**	**+0.51%**
	degenerate 01	34.92	**32.41**	**-7.19%**	1123.3	**1142.0**	**+1.66%**
	campus 00	176.64	**155.06**	**-12.22%**	411.9	**447.6**	**+8.67%**
	campus 01	303.96	**286.79**	**-5.65%**	**160.4**	153.6	**-4.24%**
	park 00	218.90	**190.86**	**-12.81%**	561.0	**591.1**	**+5.37%**
	park 01	382.10	**340.74**	**-10.82%**	281.0	**310.5**	**+10.50%**
FAST-LIVO2	Bright Screen Wall	22.61	**18.53**	**-18.05%**	2078.8	**2432.9**	**+17.03%**
	CBD Building 01	61.01	**56.34**	**-7.65%**	1734.6	**1766.6**	**+1.84%**
	CBD Building 03	206.46	**172.78**	**-16.31%**	1187.6	**1274.4**	**+7.31%**

Mapping efficiency. Table 9 shows that FormFlow-GS consistently reduces Mapping Time across M2DGR, R3LIVE, and FAST-LIVO2, covering substantially different scene scales and operating regimes. Sustained Throughput generally increases alongside this reduction, indicating that the lower mapping cost translates into higher online processing capacity. The magnitude of the throughput gain varies across sequences because system throughput also includes backbone-specific costs outside Gaussian refinement.

Quality–efficiency trade-off. Figure 9 combines the PSNR values from Table 8 with the corresponding Mapping Time above. Each arrow connects matched results on the same sequence, where movement toward the upper left indicates simultaneously higher rendering quality and lower mapping cost. Most paired operating points move toward higher PSNR and lower Mapping Time, and the macro average follows the same trend. The rendering gains therefore accompany, rather than trade against, lower online mapping cost.

C Cross-Backbone Controlled Decomposition

We further evaluate GeoBirth and CGO through a controlled 2×2 decomposition across five Gaussian mapping backbones. E00 enables neither component, E10 enables GeoBirth, E01 enables CGO, and E11 combines both.Table 10 reports rendering quality and Mapping Time under evaluation settings, allowing the formation- and execution-side effects to be examined independently.

Cross-backbone behavior. Table 10 shows that GeoBirth and CGO retain their respective roles across markedly different backbones. GeoBirth generally improves rendering quality under both native and CGO refinement, indicating that its formation benefit is not tied to a particular mapping architecture. CGO consistently reduces Mapping Time with either formation strategy, showing that its execution-side gain transfers independently of GeoBirth. Despite backbone-specific variations in individual metrics, the combined E11 setting maintains strong rendering quality with lower mapping cost, supporting the modular transfer of the formation–refinement design.

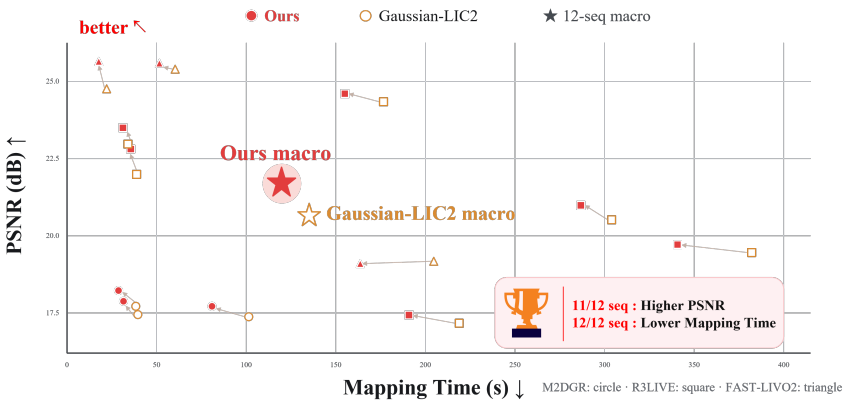

Figure 9: **Rendering quality versus mapping efficiency across 12 sequences.**

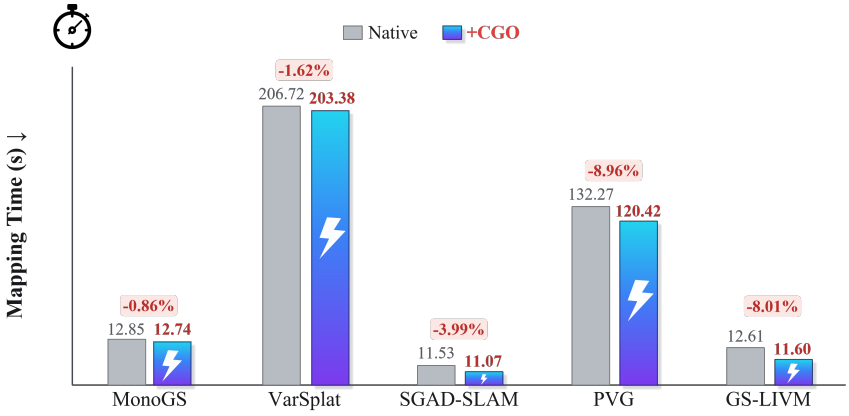

Figure 10: **Cross-backbone Mapping Time** reduction with CGO.

CGO efficiency transfer. Figure 10 isolates the execution-side effect of CGO by comparing E00 and E01 with formation fixed to Native. Mapping Time decreases across all five backbones, despite their substantially different sensing configurations and mapping pipelines. This indicates that the efficiency gain of CGO is not tied to a specific Gaussian mapping architecture.

D Qualitative Depth Geometry

Depth geometry. Figure 11 compares rendered depth under matched views.Compared with Gaussian-LIC2, our method produces cleaner depth maps around thin structures, object boundaries, and regions with rapid depth changes.These details are better separated and less blurred, showing that GeoBirth forms Gaussians that better follow the local surface geometry.

Table 10: **Controlled 2×2 decomposition across Gaussian mapping backbones.**

Backbone	Setting	PSNR↑	SSIM↑	LPIPS↓	Mapping (s)↓
MonoGS	E00	7.980	.0517	.7606	12.85
	E10	9.587	.1801	.7146	6.86
	E01	7.978	.0515	.7608	12.74
	E11	**9.613**	**.1802**	**.7137**	**6.74**
VarSplat	E00	26.508	.9078	.0901	206.72
	E10	26.609	**.9085**	**.0894**	210.14
	E01	26.507	.9078	.0902	**203.38**
	E11	**26.611**	.9081	.0896	205.44
SGAD-SLAM	E00	12.378	.4921	.3914	11.53
	E10	12.383	**.4933**	**.3847**	11.74
	E01	12.378	.4921	.3912	**11.07**
	E11	**12.483**	**.4933**	.3849	11.30
PVG	E00	24.280	.9291	.0743	132.27
	E10	24.523	.9306	.0773	122.93
	E01	**24.693**	**.9307**	.0763	120.42
	E11	24.434	.9301	**.0733**	**117.64**
GS-LIVM	E00	6.175	.3735	.7186	12.61
	E10	**6.408**	**.394**	**.7135**	12.50
	E01	6.175	.3735	.7186	**11.60**
	E11	**6.408**	**.394**	**.7135**	12.32

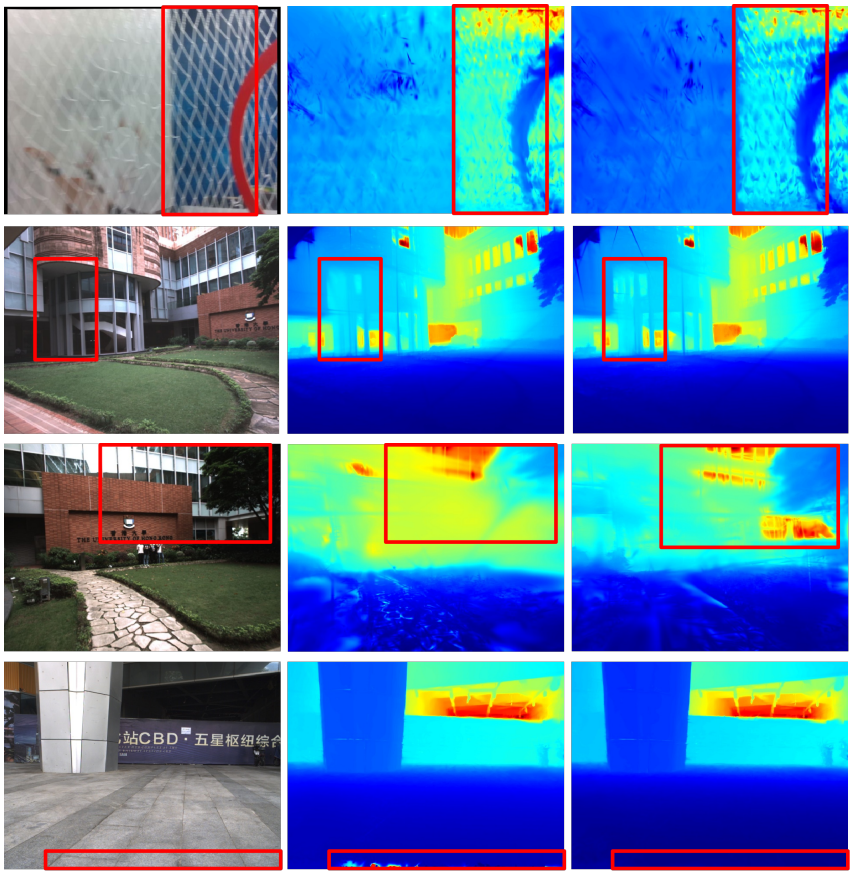

Figure 11: **Qualitative comparison of rendered depth geometry** on M2DGR (Yin et al., 2022),R3LIVE (Lin & Zhang, 2022), and FAST-LIVO2 (Zheng et al., 2025).From left to right: RGB reference, Gaussian-LIC2 depth, and our depth.Red boxes highlight thin structures, surface boundaries, and depth transitions where our method exhibits cleaner separation.

E ADDITIONAL QUALITATIVE RESULTS

We provide additional qualitative results on rendering quality, cross-backbone transfer, and GeoBirth formation behavior across different mapping systems and sensing configurations.

Additional rendering comparisons. Figure 12 compares Gaussian-LIC2 and our method on R3LIVE and FAST-LIVO2, including weakly textured regions, illumination variation, and fine geometric structures. Our method preserves clearer local texture and sharper structural boundaries, complementing the quantitative rendering results.

Cross-backbone qualitative transfer. Figure 13 evaluates GeoBirth on LoopSplat, PVG, GS-LIVM, MonoGS, SGAD-SLAM, and VarSplat. Across these backbones, GeoBirth improves local structure, surface boundaries, and fine details under substantially different sensing and mapping settings.

Additional GeoBirth formation cases. Figure 14 further pairs the newborn Gaussian geometry produced by GeoBirth with its corresponding local rendering on GS-LIVM, PVG, MonoGS, SGAD-SLAM, and VarSplat. The examples show that more surface-aligned Gaussian formation is consistently accompanied by cleaner local reconstruction across backbones.

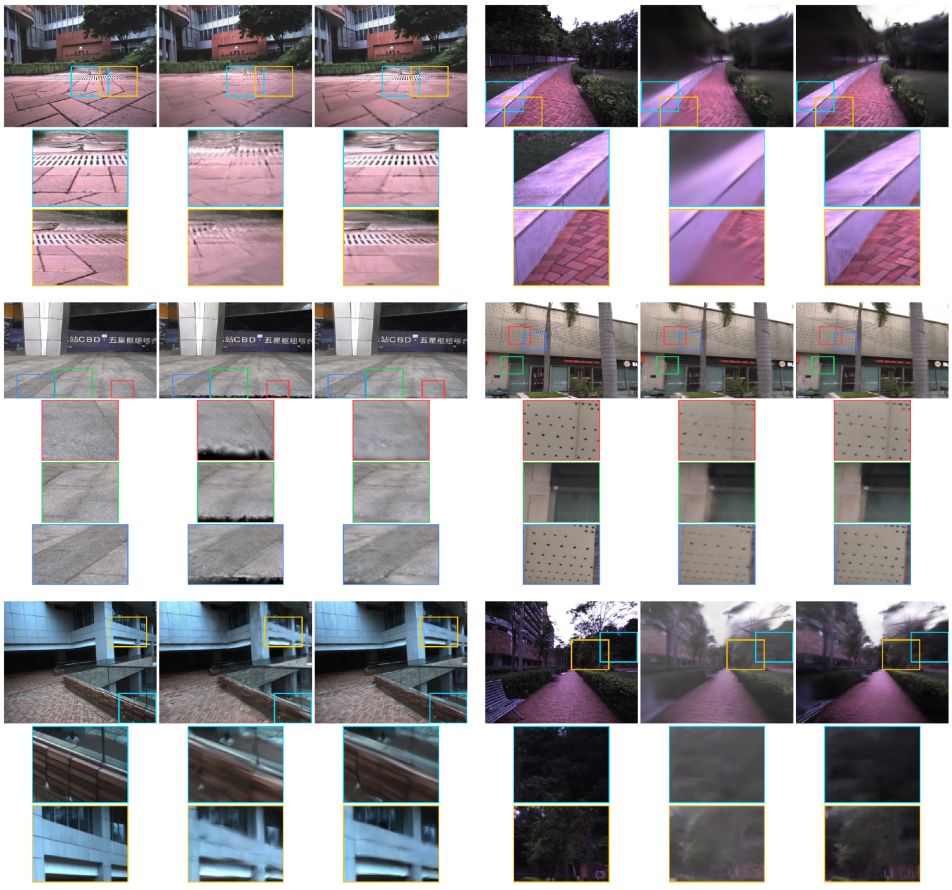

Figure 12: **Additional qualitative rendering comparisons** on R3LIVE (Lin & Zhang, 2022) and FAST-LIVO2 (Zheng et al., 2025). From left to right: RGB reference, Gaussian-LIC2, and ours.

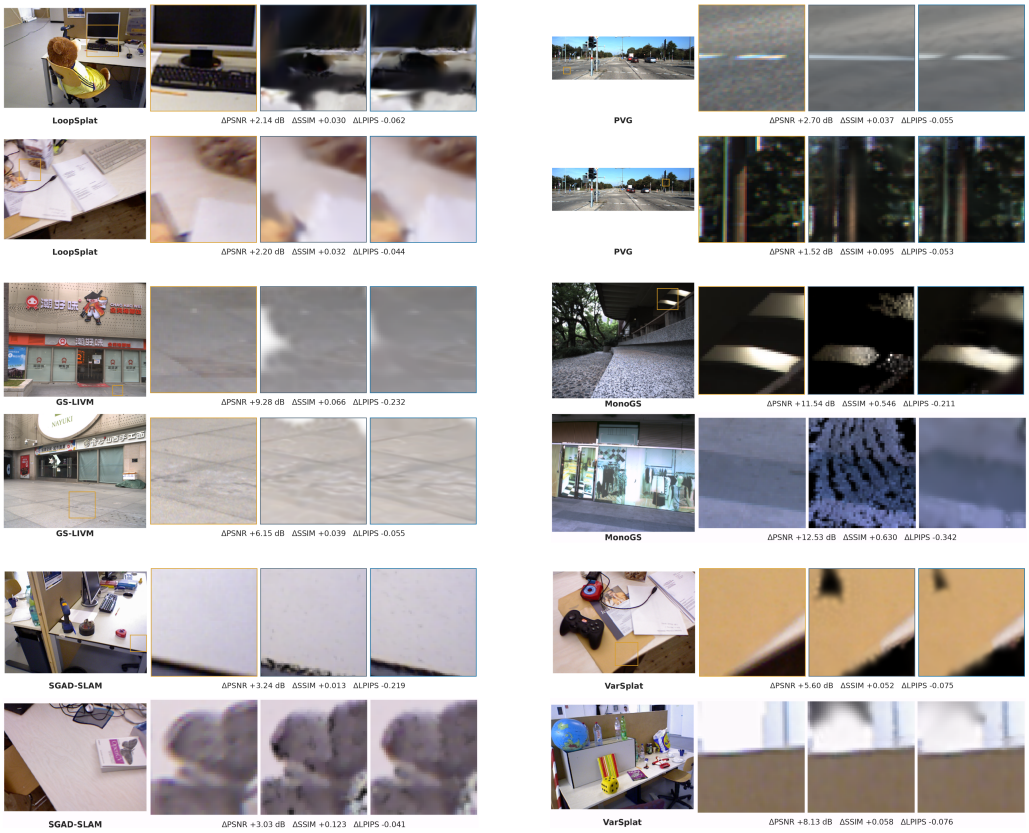

Figure 13: **Qualitative cross-backbone transfer** on R3LIVE (Lin & Zhang, 2022), FAST-LIVO2 (Zheng et al., 2025), KITTI (Geiger et al., 2012), and TUM RGB-D (Sturm et al., 2012). Each case shows an overview image and a matched local region. The three local crops are ordered as RGB reference, the native backbone reconstruction, and Ours+backbone. Reported ΔPSNR, ΔSSIM, and ΔLPIPS quantify the corresponding local improvement.

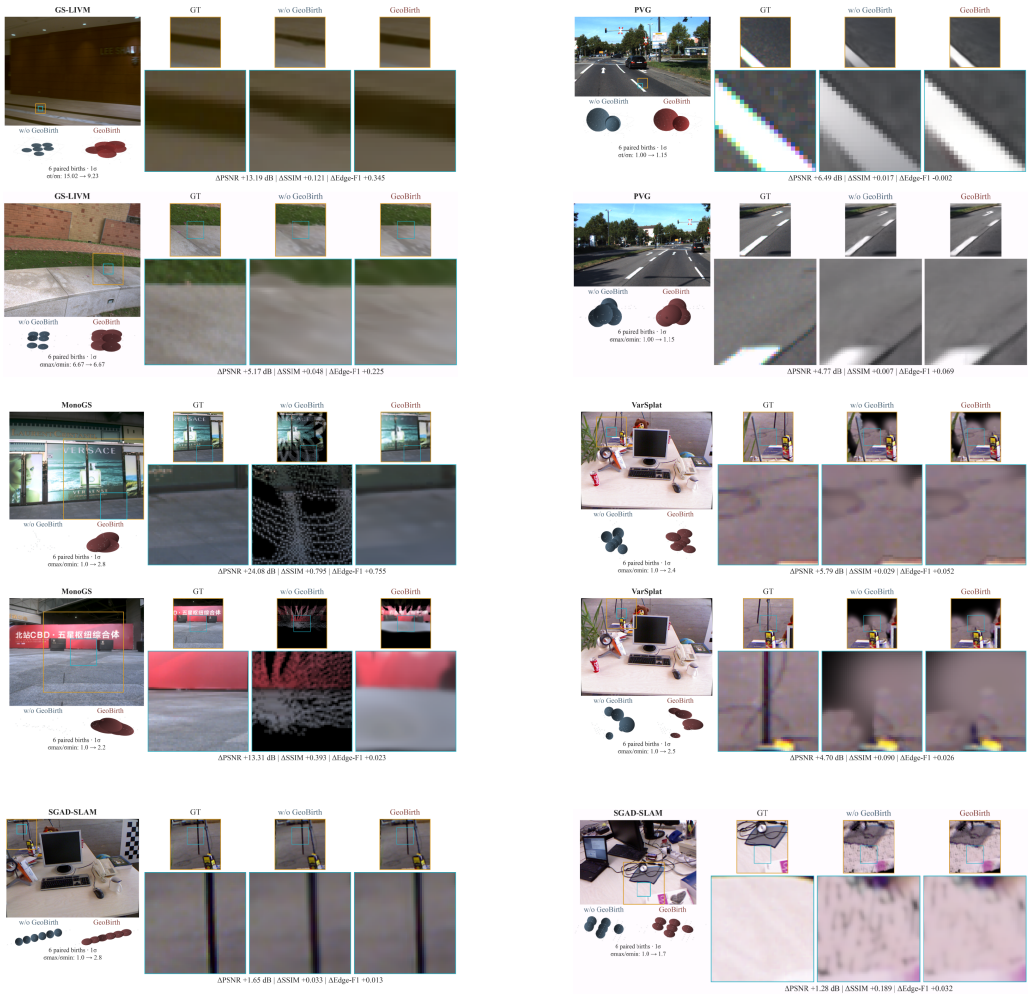

Figure 14: **Cross-backbone GeoBirth formation cases** on R3LIVE (Lin & Zhang, 2022), FAST-LIVO2 (Zheng et al., 2025), KITTI (Geiger et al., 2012), and TUM RGB-D (Sturm et al., 2012). For each case, the overview highlights the evaluated region and contrasts newborn Gaussian geometry without and with GeoBirth. The three enlarged renderings are ordered as RGB reference, w/o GeoBirth, and GeoBirth. The reported ΔPSNR, ΔSSIM, and ΔEdge-F1 measure the corresponding local rendering change.