Concord-SLAM: Cross-Render Concordance for Boundary-Native Semantic Gaussian SLAM

Danqi Lu Changxin Huang Jianqiang Li*
School of Artificial Intelligence
National Engineering Laboratory for Big Data System Computing Technology
Intelligent Computing Center
Shenzhen University

Abstract

Three-dimensional Gaussian Splatting (3DGS) has recently become a promising representation for dense SLAM, due to its explicit scene parameterization, efficient differentiable rendering, and high-quality visual reconstruction. Extending 3DGS to semantic SLAM enables the construction of maps that are not only geometrically accurate but also semantically interpretable. However, existing semantic Gaussian SLAM methods usually optimize semantic, depth, and normal renderings through separate supervision objectives. Such a design underexplores the fact that these outputs are generated from the same Gaussian field and should therefore provide a concordant explanation of the underlying scene structure. This limitation is especially problematic around object boundaries and geometric discontinuities, where inaccurate Gaussian growth can cause cross-boundary expansion, boundary blurring, and semantic leakage. To address this problem, we propose Concord-SLAM, a boundary-native semantic Gaussian SLAM framework that establishes concordance among semantic, depth, and normal renderings while promoting boundary-aware Gaussian growth. Our framework contains two key components. First, Cross-Render Concordance (CRC) jointly examines semantic, depth, and normal views rendered from the same Gaussian map, and converts their structural discrepancies in boundary responses, regional continuity, and local geometry into optimization signals. In this way, disagreement among different renderings is used as an intrinsic cue for improving the semantic-geometric coherence of the Gaussian field. Second, Boundary-Native Splatting (BNS) augments the standard opacity- and depth-residual-based Gaussian growth rule with an additional semantic-boundary insertion mask. It actively inserts new Gaussians in under-covered neighborhoods around semantic boundaries, encouraging primitives on different sides of a boundary to be represented more separately during map construction rather than correcting boundary mixing only through later optimization. Experiments on Replica and ScanNet demonstrate improved semantic mapping and reconstruction quality, while maintaining competitive camera tracking performance.

1 Introduction

Dense SLAM estimates camera motion while reconstructing geometrically accurate and visually faithful 3D maps, and has become a fundamental capability for robot perception, augmented reality, and 3D scene understanding Sucar et al. [2021], Zhu et al. [2022], Johari et al. [2023], Sandström et al. [2023]. Semantic SLAM further augments such maps with category- or instance-level information,

*Corresponding author: lijq@szu.edu.cn.

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

enabling scene-level reasoning about objects, regions, and layouts McCormac et al. [2017], Li et al. [2024a], Haghighi et al. [2023], Zhu et al. [2024]. Recently, 3D Gaussian Splatting (3DGS) has emerged as an effective representation for online dense SLAM, owing to its explicit Gaussian primitives, efficient differentiable rendering, local map updates, and high-quality novel-view synthesis Kerbl et al. [2023], Yan et al. [2024], Matsuki et al. [2024], Keetha et al. [2024], Li et al. [2024d], Li et al. [2026]. These properties make 3DGS particularly suitable for semantic Gaussian SLAM, where appearance, geometry, and semantics are expected to be maintained in a unified map Li et al. [2024c,b], Zhu et al. [2025b], Li et al. [2025], Lu et al. [2026], Yang et al. [2025], Ji et al. [2024]. However, introducing semantics into Gaussian SLAM is not simply a matter of attaching semantic attributes to Gaussians or adding an extra semantic loss. Existing semantic Gaussian SLAM methods often supervise appearance, depth, semantics, and normals through separate branches or independent objectives Li et al. [2024c,b], Zhu et al. [2025b], Lu et al. [2026]. Although such supervision uses multiple signals, it underexplores a key property of Gaussian SLAM: semantic maps, depth maps, and normal fields are different renderings of the same Gaussian scene representation. Therefore, they should not only match their own supervision targets, but also provide a structurally consistent explanation of the same scene.

Our key observation is that, in a faithful semantic Gaussian map, semantic transitions, depth discontinuities, and normal variations should be spatially aligned around object boundaries and surface changes, while they should vary coherently in smooth regions. When these renderings disagree in boundary locations, regional continuity, or local geometry, the disagreement often indicates defects in the Gaussian field, such as inaccurate primitive placement, cross-boundary expansion, insufficient boundary modeling, or ambiguous semantic assignment. Since this discrepancy is generated by the representation itself, it can serve as an intrinsic signal for map refinement without requiring additional annotations. This perspective suggests that semantic Gaussian SLAM requires more than conventional multimodal fusion. Feature concatenation, cross-modal interaction, or decoder-level aggregation can combine cues, but they do not explicitly verify whether different renderings of the same Gaussian map give a concordant structural explanation Li et al. [2024c,b], Zhu et al. [2025b], Lu et al. [2026], Yang et al. [2025]. This issue is especially important for explicit Gaussian representations, because primitives that grow across object boundaries or mixed surfaces during early mapping may cause boundary blurring and semantic leakage that later semantic supervision cannot fully remove. Effective semantic Gaussian SLAM should therefore combine rendering-level concordance constraints with representation-level boundary-aware Gaussian growth.

Motivated by this, we propose Concord-SLAM: Cross-Render Concordance for Boundary-Native Semantic Gaussian SLAM. Concord-SLAM uses inconsistencies among semantic, depth, and normal renderings as endogenous cues for correcting the Gaussian map, while encouraging Gaussian primitives to respect physical and semantic discontinuities during map construction. It contains two complementary components. Cross-Render Concordance (CRC) models structural agreement among semantic, depth, and normal renderings, and converts cross-render discrepancies into optimization signals for semantic-geometric refinement. Boundary-Native Splatting (BNS) augments opacity- and depth-residual-based Gaussian growth with semantic-boundary-aware insertion, allowing primitives to better conform to object boundaries during map construction. Together, CRC and BNS constrain Concord-SLAM at both the rendering-optimization level and the representation-growth level, enabling semantic Gaussian maps with accurate geometry, faithful appearance, clearer boundaries, and more stable semantics. The main contributions of this paper are summarized as follows:

- We propose Concord-SLAM, a boundary-native semantic Gaussian SLAM framework that treats semantic, depth, and normal renderings as coupled evidence of a shared Gaussian map, and uses their disagreement as an intrinsic signal for representation refinement.
- We introduce Cross-Render Concordance (CRC), which models structural agreement among semantic boundaries, depth discontinuities, and normal variations, and converts cross-render discrepancies into optimization constraints.
- We design Boundary-Native Splatting (BNS), which augments opacity- and depth-residual-based Gaussian growth with semantic-boundary-aware insertion, allowing primitives to better conform to object boundaries during map construction.
- We evaluate Concord-SLAM on Replica and ScanNet, demonstrating improvements in camera tracking, scene reconstruction, and semantic mapping.

2 Related Work

Gaussian SLAM 3D Gaussian Splatting has been increasingly adopted in SLAM due to its explicit primitive-based representation, differentiable splatting, and efficient rendering, making it suitable for camera tracking, dense mapping, and novel-view synthesis Kerbl et al. [2023], Yan et al. [2024], Matsuki et al. [2024], Keetha et al. [2024], Cheng et al. [2025], Wen et al. [2025], Zhu et al. [2025a]. GS-SLAM introduces 3D Gaussians as a dense visual SLAM map representation, while Gaussian Splatting SLAM further explores Gaussian-based mapping for monocular SLAM and RGB-D SLAM Yan et al. [2024], Matsuki et al. [2024]. SplaTAM jointly optimizes camera poses and Gaussian maps for RGB-D dense SLAM, showing the effectiveness of splat-based scene representations for online tracking and mapping Keetha et al. [2024]. For monocular or RGB-only settings, MonoGS++ improves Gaussian SLAM with sparse point initialization, dynamic insertion, densification, and planar regularization, further demonstrating the potential of Gaussian representations for efficient monocular RGB SLAM Li et al. [2024d]. Compared with earlier neural dense SLAM systems based on implicit fields or point-based representations, Gaussian SLAM methods benefit from explicit primitives and efficient differentiable rendering, which are more suitable for real-time map optimization and view synthesis Sucar et al. [2021], Zhu et al. [2022], Johari et al. [2023], Sandström et al. [2023], Kerbl et al. [2023]. Overall, these methods demonstrate the effectiveness of 3DGS for real-time tracking, rendering, and geometric reconstruction, but they mainly focus on geometry and appearance, with limited attention to semantic understanding and semantic boundary modeling Yan et al. [2024], Matsuki et al. [2024], Keetha et al. [2024], Li et al. [2024d].

Semantic Gaussian SLAM Semantic Gaussian SLAM associates Gaussian primitives with semantic labels, semantic features, or category-aware representations, enabling 3D maps to encode scene semantics in addition to geometry and appearance Li et al. [2024c,b], Zhu et al. [2025b], Li et al. [2025], Lu et al. [2026], Yang et al. [2025]. SGS-SLAM incorporates appearance, geometry, and semantic features into a multi-channel optimization framework, while GS^{3}LAM formulates RGB, depth, and semantic observations within a unified semantic Gaussian representation for dense semantic mapping Li et al. [2024c,b]. GS^{3}LAM further introduces semantic Gaussian splatting for semantic SLAM, while SemGauss-SLAM embeds semantic information into Gaussian primitives for unified tracking, semantic mapping, and high-quality rendering Li et al. [2024b], Zhu et al. [2025b]. Hier-SLAM scales semantic Gaussian SLAM with hierarchically categorical Gaussian splatting, and GSFF-SLAM constructs a semantic Gaussian representation through feature fields to improve semantic mapping capability Li et al. [2025], Lu et al. [2026]. OpenGS-SLAM extends semantic Gaussian SLAM to open-set dense semantic mapping by combining 3D Gaussian Splatting with object-level scene understanding Yang et al. [2025]. Although these methods benefit from explicit and optimizable Gaussian maps, they usually treat color, depth, semantics, or related geometric cues as relatively separate supervision signals, leaving the structural coupling among renderings from the same Gaussian field underexplored Li et al. [2024c,b], Zhu et al. [2025b], Li et al. [2025], Lu et al. [2026], Yang et al. [2025]. Moreover, existing Gaussian growth strategies often lack explicit awareness of semantic and geometric boundaries, which may cause cross-boundary expansion, blurred object borders, and semantic leakage Li et al. [2024c,b], Zhu et al. [2025b], Yang et al. [2025]. In contrast, Concord-SLAM exploits concordance among semantic, depth, and normal renderings from the same Gaussian field, and introduces boundary-native Gaussian allocation to construct semantic Gaussian maps with clearer boundaries and stronger structural coherence.

3 Method

Given RGB-D semantic observations $\{I_t^*, D_t^*, Y_t^*\}_{t=1}^T$, where I_t^*, D_t^*, and Y_t^* denote RGB, depth, and semantic observations, Concord-SLAM maintains an explicit semantic Gaussian map $\mathcal{G}$ for geometry, appearance, and semantics. To this end, we introduce two complementary mechanisms. Cross-Render Concordance (CRC) uses rendered depth and depth-derived normals as structural evidence to correct semantic prediction. Boundary-Native Splatting (BNS) inserts additional Gaussians near under-covered semantic boundaries, encouraging the two sides of a boundary to be represented by separable primitives during map growth. The overall pipeline of Concord-SLAM is illustrated in Figure 1.

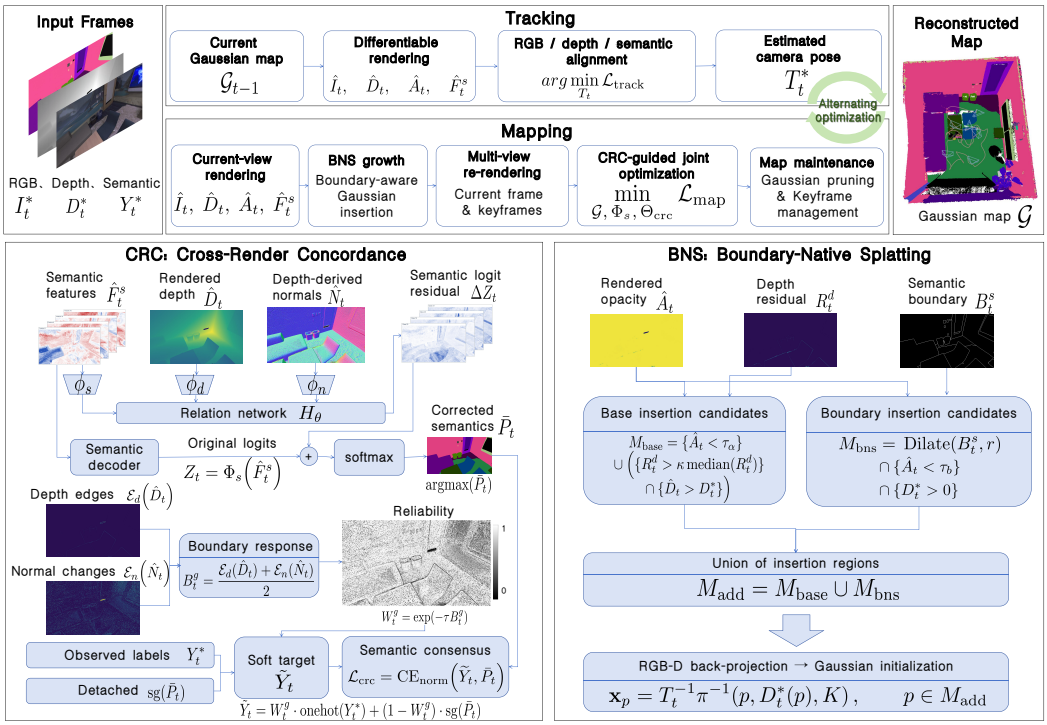

Figure 1: Overall pipeline of Concord-SLAM. The system maintains a semantic Gaussian map and jointly performs tracking, mapping, and rendering. CRC enforces cross-render concordance among semantic, depth, and normal outputs, while BNS inserts boundary-native Gaussians to improve semantic boundary representation.

3.1 Semantic Gaussian Map Representation

Concord-SLAM represents the scene as a set of optimizable semantic Gaussian primitives:

$$\mathcal{G} = \{g_i\}_{i=1}^{N}, \quad g_i = \{\boldsymbol{\mu}_i, \mathbf{s}_i, \mathbf{q}_i, \alpha_i, \mathbf{c}_i, \mathbf{f}_i^s\}. \tag{1}$$

Here, $\boldsymbol{\mu}_i$ denotes the Gaussian center, $\mathbf{s}_i$ denotes the scale, $\mathbf{q}_i$ denotes the rotation, α_i denotes the opacity, $\mathbf{c}_i$ denotes the color feature, and $\mathbf{f}_i^s$ denotes the learnable semantic feature. The semantic feature enables the same Gaussian map to render RGB, depth, and semantic predictions. Given the current camera pose T_t, the differentiable Gaussian renderer produces RGB, depth, opacity, and semantic feature renderings:

$$\hat{I}_t, \hat{D}_t, \hat{A}_t, \hat{F}_t^s = \mathcal{R}(\mathcal{G}, T_t). \tag{2}$$

Here, $\hat{I}_t$, $\hat{D}_t$, $\hat{A}_t$, and $\hat{F}_t^s$ denote the rendered RGB image, rendered depth map, rendered opacity map, and rendered semantic feature map, respectively. The rendered semantic feature is decoded into an initial semantic probability map:

$$\hat{P}_t = \mathrm{softmax}\left(\Phi_s(\hat{F}_t^s)\right), \tag{3}$$

where Φ_s denotes the semantic decoder. Concord-SLAM further derives a local normal map from the rendered depth:

$$\hat{N}_t = \Psi(\hat{D}_t, K), \tag{4}$$

where K denotes the camera intrinsics and $\Psi(\cdot)$ computes depth-derived normals by back-projection and local finite differences. Thus, semantic, depth, and normal outputs are coupled renderings of the same Gaussian map.

3.2 CRC: Cross-Render Concordance

CRC couples semantic, depth, and normal renderings by using rendered geometry as evidence to predict a residual correction for semantic logits. Given the rendered semantic feature $\hat{F}_t^s$, rendered depth $\hat{D}_t$, and depth-derived normal $\hat{N}_t$, CRC predicts the semantic logit residual as

$$\begin{aligned} \Delta Z_t &= H_\theta \left(\phi_s(\hat{F}_t^s), \phi_d(\hat{D}_t), \phi_n(\hat{N}_t) \right), \\ \bar{P}_t &= \mathrm{softmax} \left(\Phi_s(\hat{F}_t^s) + \Delta Z_t \right). \end{aligned} \tag{5}$$

Here, ϕ_s, ϕ_d, and ϕ_n denote lightweight encoders for semantic features, depth, and normals, respectively, and H_θ denotes a relation network. The residual ΔZ_t does not generate a new semantic prediction from scratch. Instead, it performs structure-guided correction on the semantic logits decoded from the Gaussian map, allowing depth and normal structures to participate in semantic optimization while keeping the semantic output tied to the map representation. CRC further adjusts semantic supervision according to local geometric reliability. Depth edges and normal variations are combined into a geometry boundary response:

$$B_t^g = \frac{1}{2} \left(\mathcal{E}_d(\hat{D}_t) + \mathcal{E}_n(\hat{N}_t) \right), \tag{6}$$

where $E_d(\cdot)$ and $E_n(\cdot)$ denote the first-order depth-gradient response and adjacent-normal variation response, respectively. Their implementation details are provided in Appendix B.10. The geometry reliability weight is then defined as

$$W_t^g = \exp(-\tau B_t^g), \tag{7}$$

where τ controls the decay rate of the reliability weight near geometric discontinuities. We use $\tau = 8.0$ for all scenes. A small B_t^g indicates a geometrically smooth region, where the semantic observation is usually reliable. A large B_t^g indicates an object boundary, occlusion boundary, or depth discontinuity, where hard semantic labels may be spatially misaligned. Based on this weight, CRC constructs a soft semantic target:

$$\tilde{Y}_t = W_t^g \cdot \mathrm{onehot}(Y_t^*) + (1 - W_t^g) \cdot \mathrm{sg}(\bar{P}_t), \tag{8}$$

where $\mathrm{sg}(\cdot)$ denotes the stop-gradient operation. In geometrically stable regions, the target is closer to the semantic observation Y_t^*, providing stronger direct supervision. Near geometric discontinuities, the influence of potentially misaligned hard labels is reduced, and the structure-corrected prediction is used as a stop-gradient soft target. The CRC loss is defined as the geometry-aware semantic consensus loss:

$$\mathcal{L}_{\mathrm{crc}} = -\frac{1}{HW \log C_t} \sum_{p \in \Omega_{\mathrm{img}}} \widetilde{Y}_t(p)^\top \log \bar{P}_t(p), \tag{9}$$

where Ω_{img} contains all $H \times W$ image pixels and C_t is a frame-dependent normalization parameter defined in Appendix B.6.

3.3 BNS: Boundary-Native Splatting

CRC improves semantic consistency during rendering and optimization, whereas BNS addresses boundary ambiguity at the Gaussian growth stage. Since 3D Gaussians are explicit spatial primitives, a primitive inserted across an object boundary may be forced to explain different semantic or geometric regions simultaneously. BNS mitigates this problem by placing additional primitives near under-covered semantic boundaries, so that different sides of a boundary can be represented more separately. Concord-SLAM first follows the common new-Gaussian insertion strategy in Gaussian SLAM to detect regions insufficiently represented by the current map. The depth residual map is defined as

$$R_t^d(p) = \left| \hat{D}_t(p) - D_t^*(p) \right| \mathbf{1}[D_t^*(p) > 0] \tag{10}$$

The base insertion region is defined as

$$M_{\mathrm{base}} = \{ \hat{A}_t < \tau_\alpha \} \cup \left\{ R_t^d > \kappa \cdot \mathrm{median}(R_t^d),\ \hat{D}_t > D_t^* \right\}, \quad \kappa = 50. \tag{11}$$

Here, τ_α is the opacity threshold, and $\kappa \cdot \mathrm{median}(R_t^d)$ is the adaptive depth residual threshold. The opacity condition detects under-covered regions, while the depth condition detects missing foreground geometry. When $\hat{D}_t > D_t^*$, the rendered surface lies behind the observed surface, indicating that the current map fails to explain the visible foreground structure. Compared with a fixed depth threshold, the adaptive residual threshold reduces sensitivity to scene scale and depth noise. However, M_{base} only detects where Gaussians are missing. For semantic mapping, boundary regions may also require more separable primitive representations, even when they are partially covered. If the two sides of a boundary are jointly explained by the same group of Gaussians, the semantic boundary can be smoothed out during optimization. Therefore, BNS further introduces a semantic-boundary- driven insertion strategy. Given the semantic observation Y_t^*, a pixel p is regarded as a semantic boundary pixel if one of its neighboring pixels has a different semantic label:

$$B_t^s(p) = \mathbf{1}\left[\exists q \in \mathcal{N}(p), Y_t^*(p) \neq Y_t^*(q)\right], \tag{12}$$

where $\mathcal{N}(p)$ denotes the neighborhood of p. To tolerate projection errors, depth noise, and semantic misalignment, BNS applies a small dilation to the semantic boundary map and inserts additional Gaussians only at under-covered boundary locations:

$$M_{\mathrm{bns}} = \mathrm{Dilate}(B_t^s, r) \cap \{\hat{A}_t < \tau_b\} \cap \{D_t^* > 0\}. \tag{13}$$

Here, r is the boundary dilation radius, and τ_b is the opacity threshold for the boundary region. BNS does not insert Gaussians at all semantic boundaries indiscriminately. It supplements primitives only where boundary neighborhoods remain insufficiently covered. The final insertion region is

$$M_{\mathrm{add}} = M_{\mathrm{base}} \cup M_{\mathrm{bns}}. \tag{14}$$

Here, M_{base} follows the standard opacity- and depth-residual-based insertion rule, while M_{bns} specifically captures under-covered semantic-boundary neighborhoods. Therefore, BNS does not replace the conventional Gaussian growth strategy. Instead, it augments it with semantic-boundary-aware insertion so that boundary regions can receive additional separable primitives when their opacity coverage is still insufficient. For each pixel $p \in M_{\mathrm{add}}$, Concord-SLAM back-projects the observed depth into 3D space and initializes the center of a new Gaussian:

$$\mathbf{x}_p = T_t^{-1}\pi^{-1}(p, D_t^*(p), K). \tag{15}$$

Here, $\pi^{-1}(\cdot)$ denotes the back-projection operation from pixel coordinates and depth to a 3D point in the camera coordinate system, and T_t^{-1} transforms the point into the world coordinate system. The color of the new Gaussian is initialized from the current RGB observation, and its scale is estimated according to the depth value and camera intrinsics. Its semantic feature is initialized as a learnable vector and optimized in subsequent mapping steps through semantic supervision and CRC constraints. BNS is not an additional differentiable loss term, but a semantic-boundary-aware Gaussian allocation strategy. Standard densification mainly considers whether a region is sufficiently covered, while BNS further considers whether semantic boundaries need more separable primitives. By inserting new Gaussians near under-covered semantic boundaries, BNS reduces the probability that a single Gaussian spans multiple semantic regions, thereby alleviating semantic mixing and boundary blurring. The online tracking-and-mapping pipeline and the complete optimization objective are provided in Appendix A.

4 Experiments

4.1 Experimental Setup

We evaluate Concord-SLAM on two standard indoor RGB-D datasets, Replica [Straub et al., 2019] and ScanNet [Dai et al., 2017]. Replica is used for evaluating localization, reconstruction, rendering, and semantic mapping under synthetic indoor scenes, while ScanNet is used to further test localization robustness in real-world RGB-D sequences. Following common protocols, we use eight Replica scenes, including Room0, Room1, Room2, Office0, Office1, Office2, Office3, and Office4, denoted as R0, R1, R2, O0, O1, O2, O3, and O4. For ScanNet, we evaluate on six sequences, including 0000, 0059, 0106, 0169, 0181, and 0207. We compare Concord-SLAM with representative Gaussian SLAM and semantic Gaussian SLAM methods, including Gaussian-SLAM [Matsuki et al., 2024], GS-SLAM [Yan et al., 2024], SplaTAM [Keetha et al., 2024], SGS-SLAM [Li et al., 2024c], GS^{3}LAM [Li

Table 1: Localization and rendering results on the Replica dataset. ATE RMSE and Depth L1 are measured in centimeters. Higher values are better for PSNR and SSIM, while lower values are better for ATE, Depth L1, and LPIPS. The best result is highlighted in **blue and bolded**, the second-best result is highlighted in green, and the third-best result is highlighted in sand.

Method	ATE RMSE ↓									Depth L1 ↓	PSNR ↑	SSIM ↑	LPIPS ↓
	R0	R1	R2	O0	O1	O2	O3	O4	Avg.				
Gaussian-SLAM	0.44	0.32	0.31	0.44	0.52	**0.23**	0.17	2.25	0.58	1.50	38.94	0.968	0.070
GS-SLAM	0.48	0.53	0.33	0.52	0.41	0.59	0.46	0.70	0.50	1.16	34.27	0.975	0.082
SplaTAM	0.31	0.40	0.29	0.47	0.27	0.29	0.32	0.55	0.36	0.53	34.11	0.970	0.100
SGS-SLAM	0.46	0.45	0.29	0.46	0.23	0.45	0.42	0.55	0.41	0.36	34.66	0.973	0.096
GS^{3}LAM	0.27	0.25	0.28	0.67	0.21	0.33	0.30	0.65	0.37	0.87	36.26	**0.989**	0.052
SemGauss-SLAM	**0.26**	0.42	**0.27**	0.34	**0.17**	0.32	0.36	**0.49**	0.33	0.50	35.03	0.982	0.062
GSFF-SLAM	0.34	0.29	0.33	0.35	0.24	0.25	**0.15**	0.54	0.31	0.72	38.67	0.974	**0.035**
Ours	0.27	**0.22**	**0.27**	**0.33**	0.24	0.32	0.25	0.50	**0.30**	**0.35**	**39.57**	**0.989**	0.060

Table 2: Localization results on the ScanNet dataset. ATE RMSE is measured in centimeters, and lower values indicate better performance. The best result is highlighted in **blue and bolded**, the second-best result is highlighted in green, and the third-best result is highlighted in sand.

Method	0000	0059	0106	0169	0181	0207	Avg.
SGS-SLAM	11.15	9.54	**10.43**	10.70	11.28	6.11	**9.87**
SemGauss-SLAM	11.87	7.97	14.60	**8.70**	**9.78**	8.97	10.32
GSFF-SLAM	**9.60**	7.80	16.80	9.60	21.90	**5.60**	11.88
Ours	10.77	**7.53**	12.75	10.42	11.60	7.94	10.17

et al., 2024b], SemGauss-SLAM [Zhu et al., 2025b], and GSFF-SLAM [Lu et al., 2026]. For localization, we report ATE RMSE in centimeters. For reconstruction and rendering, we report Depth L1, PSNR, SSIM [Wang et al., 2004], and LPIPS [Zhang et al., 2018]. For semantic mapping, we use mIoU. For runtime efficiency, we report Tracking FPS, Mapping FPS, and overall SLAM FPS. All experiments are conducted in an online RGB-D semantic SLAM setting. The system processes frames sequentially without using future-frame information. The first frame initializes the Gaussian map and defines the world reference frame. Subsequent frames are processed by alternating camera tracking and map optimization. Unless otherwise specified, results reported for existing methods are taken from the original papers when available, and missing entries are reproduced under the same experimental setting. Detailed dataset descriptions, metric definitions, baseline configurations, and implementation settings are provided in Appendix B.

4.2 Experimental Results

4.2.1 Localization Performance

As shown in Table 1 and Table 2, localization accuracy reflects whether the reconstructed map can provide stable constraints for camera tracking. On Replica, the proposed method achieves the best average ATE of 0.30, outperforming GSFF-SLAM with 0.31, SemGauss-SLAM with 0.33, SplaTAM with 0.36, and GS^{3}LAM with 0.37. Although it does not obtain the lowest error in every scene, its best average result indicates stronger cross-scene stability and shows that the constructed semantic Gaussian map can support reliable tracking across different indoor environments. On ScanNet, the proposed method achieves an average ATE of 10.17, outperforming SemGauss-SLAM with 10.32 and GSFF-SLAM with 11.88, while being slightly inferior to SGS-SLAM with 9.87. This suggests that the method remains competitive on real-world RGB-D sequences, where sensor noise, missing depth values, complex motion, and trajectory drift are more prominent. Overall, the results show that the proposed semantic consistency modeling does not weaken camera tracking and can maintain robust localization performance.

Table 3: Semantic mapping and ablation results on the Replica dataset, evaluated by mIoU (%). Ours* denotes the variant without Boundary-Native Splatting (BNS), and Ours** denotes the variant without Cross-Render Concordance (CRC). Higher values indicate better performance. The best result is highlighted in **blue and bolded**, the second-best result is highlighted in green, and the third-best result is highlighted in sand.

Method	R0	R1	R2	O0	O1	O2	O3	O4	Avg.
SGS-SLAM	92.95	92.91	92.10	92.90	91.45	92.67	92.38	91.93	92.41
GS^{3}LAM	96.83	96.68	96.40	96.61	96.01	95.67	95.12	96.74	96.26
SemGauss-SLAM	96.30	95.82	96.51	96.72	96.55	96.48	96.36	95.84	96.32
GSFF-SLAM	95.20	96.90	95.98	95.12	93.44	94.96	94.74	93.97	95.04
Ours	**98.35**	**98.88**	**98.54**	**98.27**	**98.40**	**98.06**	**97.51**	**97.15**	**98.15**
Ours*(w/o BNS)	97.69	97.64	97.50	96.31	96.78	96.12	96.55	96.48	96.88
Ours**(w/o CRC)	97.54	97.41	97.74	97.64	96.70	96.47	97.11	97.04	97.21

Table 4: Runtime efficiency comparison in terms of Tracking FPS, Mapping FPS, and overall SLAM FPS. Higher values indicate better efficiency. The best result is highlighted in **blue and bolded**, the second-best result is highlighted in green, and the third-best result is highlighted in sand.

Method	Tracking FPS ↑	Mapping FPS ↑	SLAM FPS ↑
SGS-SLAM	5.27	3.52	2.11
GS^{3}LAM	2.77	1.75	1.07
GSFF-SLAM	3.77	2.78	1.60
Ours	**6.49**	**4.44**	**2.64**

4.2.2 Reconstruction Performance

Table 1 also reports reconstruction and rendering quality on Replica. The proposed method achieves the best Depth L1 of 0.35, outperforming SGS-SLAM with 0.36, SemGauss-SLAM with 0.50, SplaTAM with 0.53, and GS^{3}LAM with 0.87. This indicates that the proposed method renders more accurate depth maps and improves geometric consistency in the Gaussian representation. For appearance rendering, the proposed method obtains the highest PSNR of 39.57 and an SSIM of 0.989, tying with GS^{3}LAM for the best result. These results show that the method improves geometry and semantics without sacrificing RGB reconstruction quality. Its LPIPS is 0.060, which is better than SGS-SLAM and SemGauss-SLAM, but worse than GSFF-SLAM with 0.035 and GS^{3}LAM with 0.052. This indicates that the method still has room for improvement in perceptual similarity, especially for local textures and high-frequency appearance details.

4.2.3 Semantic Mapping Performance

Figure 2 shows that the proposed method produces semantic maps closer to the ground truth than SGS-SLAM and GS^{3}LAM. It better preserves category consistency in large planar regions and yields cleaner boundaries around objects, furniture edges, and adjacent semantic regions, indicating reduced boundary ambiguity and cross-category diffusion. The quantitative semantic mapping results are reported in Table 3. The proposed method achieves the best semantic performance in all tested Replica scenes, with an average mIoU of 98.15. It outperforms SemGauss-SLAM with 96.32, GS^{3}LAM with 96.26, GSFF-SLAM with 95.04, and SGS-SLAM with 92.41. Compared with the closest competitor SemGauss-SLAM, it improves mIoU by 1.83 percentage points; compared with GSFF-SLAM and SGS-SLAM, the improvements are 3.11 and 5.74 percentage points, respectively. These gains mainly come from the complementary effects of BNS and CRC. BNS inserts new Gaussians near under-covered semantic-boundary neighborhoods, reducing the chance that one primitive spans different categories. CRC further uses depth and normal information to regularize semantic prediction, encouraging semantic boundaries to agree with geometric structures. Therefore, the semantic improvement is not only due to stronger semantic supervision, but also to the joint consistency modeling among semantics, depth, and normals.

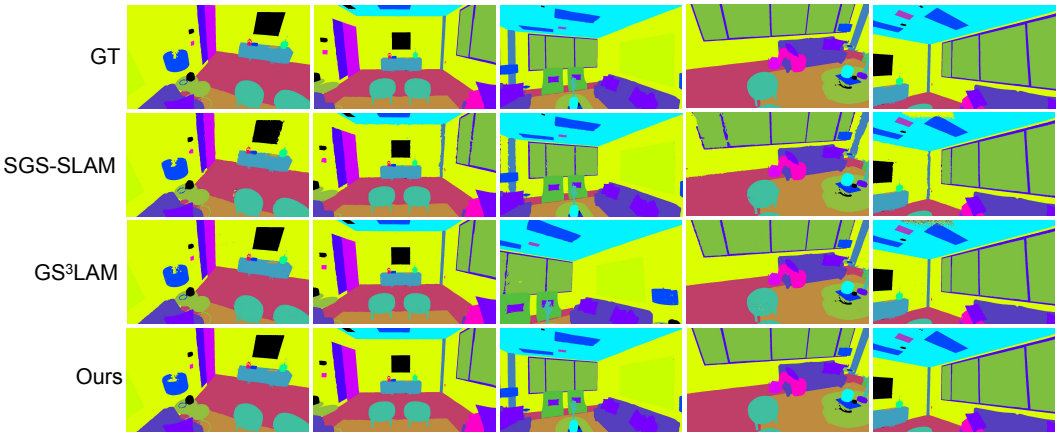

Figure 2: Qualitative comparison of semantic mapping results. Each column corresponds to a different test scene, and each row shows the semantic map produced by GT, SGS-SLAM, GS^{3}LAM, and the proposed method, respectively. The proposed method produces semantic maps closer to the ground truth, especially around object boundaries, contact regions, and fine structures.

4.2.4 Runtime Efficiency

Runtime efficiency is important for online dense semantic SLAM. As reported in Table 4, the proposed method achieves Tracking FPS, Mapping FPS, and overall SLAM FPS of 6.49, 4.44, and 2.64, respectively, all of which are the highest among the compared methods. Compared with SGS-SLAM, the proposed method improves Tracking FPS from 5.27 to 6.49, Mapping FPS from 3.52 to 4.44, and overall SLAM FPS from 2.11 to 2.64, corresponding to relative improvements of 23.1%, 26.1%, and 25.1%. These results demonstrate that Concord-SLAM achieves higher tracking, mapping, and overall SLAM throughput than the compared methods under the reported experimental setting. CRC performs lightweight residual correction on semantic logits using rendered depth and depth-derived normals as structural evidence, while BNS modifies the Gaussian growth mask only at under-covered boundary regions. Neither component requires global scene-level matching or a separate post-processing stage. As a result, Concord-SLAM preserves high online efficiency while improving semantic boundary representation and cross-render structural consistency.

4.2.5 Ablation Study

We conduct ablation experiments on Replica to evaluate the individual contributions of Boundary-Native Splatting (BNS) and Cross-Render Concordance (CRC). As shown in Table 3, Ours* denotes the variant without BNS, and Ours** denotes the variant without CRC. The full model achieves the best average semantic score of 98.15, outperforming both ablated variants. **Effect of BNS.** When BNS is removed, the average semantic score drops from 98.15 to 96.88, with a decrease of 1.27 percentage points. The degradation is particularly clear in scenes O0, O1, and O2, where the scores decrease from 98.27, 98.40, and 98.06 to 96.31, 96.78, and 96.12, respectively. These results indicate that BNS plays an important role in semantic Gaussian growth. By inserting new Gaussians around under-covered semantic-boundary neighborhoods, BNS improves boundary coverage and reduces cross-boundary semantic mixing. Without this boundary-aware growth strategy, Gaussians are more likely to spread across adjacent regions with different semantic labels, leading to blurred boundaries and less accurate semantic maps. **Effect of CRC.** When CRC is removed, the average semantic score decreases from 98.15 to 97.21, with a drop of 0.94 percentage points. This shows that CRC also contributes consistently to semantic mapping performance. Without CRC, semantic optimization relies more directly on semantic supervision and lacks explicit structural feedback from rendered depth and normal cues. As a result, inconsistent predictions around object boundaries are less effectively corrected. CRC mitigates this problem by enforcing cross-render consistency among semantic, depth, and normal outputs, allowing geometric structure to provide useful guidance for semantic refinement.

5 Conclusion

This paper presents Concord-SLAM, a self-consistent semantic Gaussian SLAM framework for dense semantic mapping. The key idea is to exploit the consistency among semantic, depth, and normal renderings from the same Gaussian field. To this end, Concord-SLAM introduces Cross-Render Concordance (CRC) to use cross-render structural inconsistencies as refinement signals, and Boundary-Native Splatting (BNS) to improve Gaussian growth around under-covered semantic-boundary regions. Experiments on Replica and ScanNet show that Concord-SLAM improves reconstruction, rendering, and semantic mapping quality, while maintaining competitive tracking performance, confirming the effectiveness of CRC and BNS.

Acknowledgments

This work is supported in part by the National Natural Science Funds for Distinguished Young Scholar (No. 62325307), in part by the National Key R&D Program of China (No. 2025YFB4712600), in part by the National Natural Science Foundation Major Scientific Research Instrument Development Project (No. 62527809), in part by the Shenzhen Key Industry R&D Program Project (No. ZDCY20250901102300001), in part by the National Natural Science Foundation of China (Nos. 62373257, 62473264, 62203134, 62403325), in part by the Natural Science Foundation of Guangdong Province (Nos. 2023B1515120038, 2026A1515011532), in part by Shenzhen Science and Technology Innovation Commission (No. KJZD20230923113801004), in part by the Key-Area Research and Development Program of Guangdong Province (No. 2025B0909020002), in part by the Outstanding Youth Project of Guangdong Basic and Applied Basic Research Foundation (No. 2023B1515020064), and in part by the Open Project of State Key Laboratory of Synthetical Automation for Process Industries (No. SAPI-2025-KFKT-11). This work is supported by the Intelligent Computing Center of Shenzhen University. The authors declare no competing interests.

References

Chong Cheng, Sicheng Yu, Zijian Wang, Yifan Zhou, and Hao Wang. Outdoor monocular SLAM with global scale-consistent 3d gaussian pointmaps. In *IEEE/CVF International Conference on Computer Vision, ICCV 2025, Honolulu, HI, USA, October 19-25, 2025*, pages 26035–26044. IEEE, 2025. doi: 10.1109/ICCV51701.2025.02415. URL `https://doi.org/10.1109/ICCV51701.2025.02415`.

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

Yasaman Haghighi, Suryansh Kumar, Jean-Philippe Thiran, and Luc Van Gool. Neural implicit dense semantic SLAM. *arXiv preprint arXiv:2304.14560*, 2023. doi: 10.48550/arXiv.2304.14560.

Yiming Ji, Yang Liu, Guanghu Xie, Boyu Ma, Zongwu Xie, and Hong Liu. NEDS-SLAM: A neural explicit dense semantic SLAM framework using 3d gaussian splatting. *IEEE Robotics Autom. Lett.*, 9(10):8778–8785, 2024. doi: 10.1109/LRA.2024.3451390. URL `https://doi.org/10.1109/LRA.2024.3451390`.

Mohammad Mahdi Johari, Camilla Carta, and François Fleuret. ESLAM: Efficient dense SLAM system based on hybrid representation of signed distance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17408–17419, 2023. doi: 10.1109/CVPR52729.2023.01670.

Nikhil Varma Keetha, Jay Karhade, Krishna Murthy Jatavallabhula, Gengshan Yang, Sebastian A. 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 (CVPR)*, pages 21357–21366, 2024. doi: 10.1109/CVPR52733.2024.02018.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4):139:1–139:14, 2023. doi: 10.1145/3592433.

Boying Li, Zhixi Cai, Yuan-Fang Li, Ian Reid, and Hamid Rezatofighi. Hier-SLAM: Scaling-up semantics in SLAM with a hierarchically categorical gaussian splatting. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 9748–9754, 2025. doi: 10.1109/ICRA55743.2025.11127775.

Guanghao Li, Kerui Ren, Linning Xu, Zhewen Zheng, Changjian Jiang, Xin Gao, Bo Dai, Jian Pu, Mulin Yu, and Jiangmiao Pang. ARTDECO: toward high-fidelity on-the-fly reconstruction with hierarchical gaussian structure and feed-forward guidance. In Carl Vondrick, Bharath Hariharan, Colin Raffel, Lerrel Pinto, Diyi Yang, and Aleksandra Faust, editors, *The Fourteenth International Conference on Learning Representations, ICLR 2026, Rio de Janeiro, Brazil, April 23-27, 2026*. proceedings.iclr.cc, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/a87f2df7c4ab0213c6ea228e7b7f0a4d-Abstract-Conference.html.

Kunyi Li, Michael Niemeyer, Nassir Navab, and Federico Tombari. DNS-SLAM: Dense neural semantic-informed SLAM. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7839–7846, 2024a. doi: 10.1109/IROS58592.2024.10803056.

Linfei Li, Lin Zhang, Zhong Wang, and Ying Shen. GS^{3}LAM: Gaussian semantic splatting SLAM. In *Proceedings of the ACM International Conference on Multimedia (ACM MM)*, pages 3019–3027, 2024b. doi: 10.1145/3664647.3680739.

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 *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 163–179, 2024c. doi: 10.1007/978-3-031-72751-1_10.

Renwu Li, Wenjing Ke, Dong Li, Lu Tian, and Emad Barsoum. MonoGS++: Fast and accurate monocular RGB gaussian SLAM. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2024d.

Zuxing Lu, Xin Yuan, Shaowen Yang, Jingyu Liu, and Changyin Sun. GSFF-SLAM: 3d semantic gaussian splatting SLAM via feature field. *Expert Systems with Applications*, 298:129601, 2026. doi: 10.1016/j.eswa.2025.129601.

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 (CVPR)*, pages 18039–18048, 2024. doi: 10.1109/CVPR52733.2024.01708.

John McCormac, Ankur Handa, Andrew J. Davison, and Stefan Leutenegger. SemanticFusion: Dense 3d semantic mapping with convolutional neural networks. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 4628–4635, 2017. doi: 10.1109/ICRA.2017.7989538.

Erik Sandström, Yue Li, Luc Van Gool, and Martin R. Oswald. Point-SLAM: Dense neural point cloud-based SLAM. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 18387–18398, 2023. doi: 10.1109/ICCV51070.2023.01690.

Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J. Engel, Raul Mur-Artal, Carl Yuheng Ren, Shobhit Verma, Anton Clarkson, Mingfei Yan, Brian Budge, Yajie Yan, Xiaqing Pan, June Yon, Yuyang Zou, Kimberly Leon, Nigel Carter, Jesus Briales, Tyler Gillingham, Elias Mueggler, Luis Pesqueira, Manolis Savva, Dhruv Batra, Hauke M. Strasdat, Renzo De Nardi, Michael Goesele, Steven Lovegrove, and Richard A. Newcombe. The replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*, 2019.

Edgar Sucar, Shikun Liu, Joseph Ortiz, and Andrew J. Davison. iMAP: Implicit mapping and positioning in real-time. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pages 6209–6218. IEEE, 2021. doi: 10.1109/ICCV48922.2021.00617. URL https://doi.org/10.1109/ICCV48922.2021.00617.

Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Process.*, 13(4):600–612, 2004. doi: 10.1109/TIP.2003.819861. URL https://doi.org/10.1109/TIP.2003.819861.

Tianci Wen, Zhiang Liu, and Yongchun Fang. Segs-slam: Structure-enhanced 3d gaussian splatting slam with appearance embedding. In *IEEE/CVF International Conference on Computer Vision, ICCV 2025, Honolulu, HI, USA, October 19-25, 2025*, pages 28103–28113. IEEE, 2025. doi: 10.1109/ICCV51701.2025.02609. URL https://doi.org/10.1109/ICCV51701.2025.02609.

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 (CVPR)*, pages 19595–19604, 2024. doi: 10.1109/CVPR52733.2024.01853.

Dianyi Yang, Yu Gao, Xihan Wang, Yufeng Yue, Yi Yang, and Mengyin Fu. OpenGS-SLAM: Open-set dense semantic SLAM with 3d gaussian splatting for object-level scene understanding. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 8486–8492, 2025. doi: 10.1109/ICRA55743.2025.11127983.

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 Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 586–595. Computer Vision Foundation / IEEE Computer Society, 2018. doi: 10.1109/CVPR.2018.00068. URL http://openaccess.thecvf.com/content_cvpr_2018/html/Zhang_The_Unreasonable_Effectiveness_CVPR_2018_paper.html.

Liyuan Zhu, Yue Li, Erik Sandström, Shengyu Huang, Konrad Schindler, and Iro Armeni. Loopsplat: Loop closure by registering 3d gaussian splats. In *International Conference on 3D Vision, 3DV 2025, Singapore, March 25-28, 2025*, pages 156–167. IEEE, 2025a. doi: 10.1109/3DV66043.2025.00020. URL https://doi.org/10.1109/3DV66043.2025.00020.

Siting Zhu, Guangming Wang, Hermann Blum, Jiuming Liu, Liang Song, Marc Pollefeys, and Hesheng Wang. SNI-SLAM: semantic neural implicit SLAM. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pages 21167–21177. IEEE, 2024. doi: 10.1109/CVPR52733.2024.02000. URL https://doi.org/10.1109/CVPR52733.2024.02000.

Siting Zhu, Renjie Qin, Guangming Wang, Jiuming Liu, and Hesheng Wang. Semgauss-SLAM: Dense semantic gaussian splatting SLAM. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 21174–21181, 2025b. doi: 10.1109/IROS60139.2025.11246462.

Zihan Zhu, Songyou Peng, Viktor Larsson, Weiwei Xu, Hujun Bao, Zhaopeng Cui, Martin R. Oswald, and Marc Pollefeys. NICE-SLAM: Neural implicit scalable encoding for SLAM. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12776–12786, 2022. doi: 10.1109/CVPR52688.2022.01245.

Appendix

A SLAM System Framework and Optimization Details

Concord-SLAM follows an online tracking-and-mapping framework. Given an RGB-D semantic sequence $\{I_t^*, D_t^*, Y_t^*\}_{t=1}^T$, the system incrementally estimates the camera trajectory and constructs a persistent semantic Gaussian map $\mathcal{G}$. Each Gaussian primitive stores its position, scale, rotation, opacity, color feature, and learnable semantic feature. For each incoming frame, Concord-SLAM first estimates the current camera pose with the Gaussian map fixed, and then updates the semantic Gaussian map using the current observation together with historical keyframes.

A.1 Initialization

For the first frame, Concord-SLAM initializes the world coordinate system using the camera coordinate system of that frame. Each valid depth pixel is back-projected into a 3D point and used to

initialize a Gaussian primitive. The color of each Gaussian is initialized from the corresponding RGB observation, and the scale is estimated according to the depth value and camera intrinsics:

$$s_p \propto \frac{D_0^*(p)}{(f_x + f_y)/2}. \tag{16}$$

Here, $D_0^*(p)$ denotes the observed depth of pixel p in the first frame, and f_x and f_y denote the focal lengths of the camera. A learnable semantic feature is also initialized for each Gaussian primitive, so that the map can subsequently render semantic predictions in addition to RGB and depth.

A.2 Tracking

For each incoming frame t, Concord-SLAM first obtains an initial camera pose estimate using a constant-velocity motion model. Then, with the semantic Gaussian map $\mathcal{G}$ fixed, the current camera pose is optimized by minimizing a tracking objective:

$$T_t^\star = \arg\min_{T_t} \mathcal{L}_{\text{track}}. \tag{17}$$

Given a candidate pose T_t, the differentiable Gaussian renderer produces RGB, depth, opacity, and semantic renderings from the current map. The tracking loss is defined as

$$\mathcal{L}_{\text{track}} = \lambda_I \mathcal{L}_{\text{rgb}} + \lambda_D \mathcal{L}_{\text{depth}} + \lambda_S \mathcal{L}_{\text{sem}}. \tag{18}$$

Here, $\mathcal{L}_{\text{rgb}}$, $\mathcal{L}_{\text{depth}}$, and $\mathcal{L}_{\text{sem}}$ denote RGB, depth, and semantic reprojection losses, respectively. During tracking, the Gaussian map parameters are kept fixed, and only the camera pose of the current frame is optimized. To improve tracking robustness, the loss is computed only over valid depth regions and reliable mapped regions. Specifically, regions with insufficient opacity coverage are excluded because they are not yet well represented by the current Gaussian map. Pixels near semantic boundaries can also be ignored during tracking, since they are more likely to be affected by occlusion, depth discontinuities, and semantic-label misalignment. During pose optimization, the pose with the lowest tracking loss is recorded as the final pose estimate for the current frame.

A.3 Gaussian Growth and Boundary-Native Splatting

After estimating the current camera pose, Concord-SLAM updates the semantic Gaussian map. Before continuous optimization, the system determines where new Gaussian primitives should be inserted. The base growth strategy follows common Gaussian SLAM practice and detects regions insufficiently represented by the current map according to opacity coverage and depth residuals. Low rendered opacity indicates that the region is not sufficiently covered by existing Gaussians. A large depth residual, especially when the rendered depth lies behind the observed depth, indicates that the current map fails to explain visible foreground geometry. Boundary-Native Splatting (BNS) further extends this strategy by considering semantic boundaries. Instead of inserting new Gaussians only in globally under-covered regions, BNS gives additional attention to under-covered neighborhoods around semantic boundaries. This design encourages different sides of a semantic boundary to be represented by more separable Gaussian primitives. Therefore, BNS reduces the chance that a single Gaussian primitive spans multiple semantic regions, which helps alleviate boundary blurring and semantic mixing from the map growth stage. The final insertion mask is obtained by combining the base insertion region and the BNS boundary insertion region. For each selected pixel, the system back-projects the pixel into 3D using the observed depth and initializes a new Gaussian primitive. Its position is initialized by back-projection, its color is initialized from the RGB observation, and its semantic feature is learned during subsequent mapping optimization.

A.4 Mapping Optimization

After Gaussian insertion, Concord-SLAM optimizes the semantic Gaussian map using the current frame and historical keyframes. The optimized variables include Gaussian centers, scales, rotations, opacities, color features, and semantic features:

$$\{\boldsymbol{\mu}_i, \mathbf{s}_i, \mathbf{q}_i, \alpha_i, \mathbf{c}_i, \mathbf{f}_i^s\}_{i=1}^N. \tag{19}$$

The mapping stage jointly optimizes the Gaussian map, the semantic decoder, and the CRC module:

$$\mathcal{G}^\star, \Phi_s^\star, \Theta_{\text{crc}}^\star = \arg\min_{\mathcal{G}, \Phi_s, \Theta_{\text{crc}}} \mathcal{L}_{\text{map}}. \tag{20}$$

Here, Φ_s denotes the semantic decoder, and Θ_{crc} denotes the parameters of the Cross-Render Concordance module. The mapping objective is defined as

$$\mathcal{L}_{\mathrm{map}} = \lambda_I \mathcal{L}_{\mathrm{rgb}} + \lambda_D \mathcal{L}_{\mathrm{depth}} + \lambda_S \mathcal{L}_{\mathrm{sem}} + \lambda_C \mathcal{L}_{\mathrm{crc}} + \lambda_R \mathcal{L}_{\mathrm{reg}}. \tag{21}$$

Here, $\mathcal{L}_{\mathrm{rgb}}$ denotes the RGB reconstruction loss, $\mathcal{L}_{\mathrm{depth}}$ denotes the depth reconstruction loss, $\mathcal{L}_{\mathrm{sem}}$ denotes the semantic supervision loss, $\mathcal{L}_{\mathrm{crc}}$ denotes the Cross-Render Concordance loss, and $\mathcal{L}_{\mathrm{reg}}$ denotes regularization terms for stabilizing Gaussian geometry, opacity, and scale. The RGB loss combines pixel-wise reconstruction and structural similarity terms. The depth loss is computed over valid depth regions. The semantic loss is computed between the semantic labels and the logits decoded from the rendered semantic features. The CRC loss further constrains semantic prediction using geometric evidence from rendered depth and depth-derived normals. In this way, the semantic, depth, and normal renderings from the same Gaussian field are encouraged to provide structurally consistent explanations of the scene.

A.5 Roles of CRC and BNS

CRC and BNS affect Concord-SLAM in different but complementary ways. CRC acts during differentiable optimization. It mainly optimizes the semantic features of Gaussian primitives, the semantic decoder, and the cross-render residual network through the mapping loss. By using rendered depth and depth-derived normals as structural evidence, CRC improves semantic prediction and encourages semantic boundaries to be consistent with geometric discontinuities. BNS acts during map growth. It does not introduce an additional differentiable loss term. Instead, it changes the spatial allocation of newly inserted Gaussian primitives by placing additional Gaussians near under-covered semantic-boundary regions. This makes the explicit Gaussian representation more boundary-aware before subsequent optimization. Therefore, BNS improves boundary separability at the representation-growth level, while CRC improves cross-render structural consistency at the rendering-optimization level. Together, they enable Concord-SLAM to construct a dense semantic Gaussian map with clearer boundaries, more stable semantics, and more coherent geometry.

A.6 Keyframe Maintenance

Concord-SLAM stores keyframes at fixed frame intervals. Each keyframe contains the RGB image, depth map, semantic labels, and the estimated camera pose. During mapping optimization, the system alternates between the current frame and randomly sampled historical keyframes. This replay-based strategy allows the map to absorb new observations while preserving consistency with previous views. To control map size and avoid redundant primitives, the system periodically prunes low-opacity or abnormal Gaussians and regularizes Gaussian scales. Overall, Concord-SLAM follows an online pipeline consisting of pose estimation, boundary-aware Gaussian growth, map optimization, and keyframe maintenance. Tracking aligns the current frame to the existing Gaussian map, while mapping constructs a structurally self-consistent and boundary-aware semantic Gaussian field through BNS and CRC.

B Additional Experimental Details

B.1 Datasets and Evaluation Protocol

We evaluate Concord-SLAM on two indoor RGB-D datasets, Replica [Straub et al., 2019] and ScanNet [Dai et al., 2017]. Replica is a synthetic indoor dataset with high-quality RGB images, depth maps, semantic annotations, and ground-truth camera trajectories. It provides clean geometry and accurate camera poses, and is therefore suitable for evaluating dense SLAM systems in terms of localization, reconstruction, rendering, and semantic mapping. Following common practice in dense RGB-D SLAM and Gaussian SLAM, we conduct experiments on eight representative Replica scenes: Room0, Room1, Room2, Office0, Office1, Office2, Office3, and Office4. These scenes are denoted as R0, R1, R2, O0, O1, O2, O3, and O4, respectively. ScanNet is a real-world indoor RGB-D dataset. Compared with Replica, ScanNet contains stronger sensor noise, missing depth values, occlusions, motion blur, and less accurate camera trajectories. Therefore, it is used to evaluate the robustness and generalization ability of the proposed method in more realistic indoor environments. Following the evaluation setting of related Gaussian SLAM methods, we select six ScanNet sequences, including

0000, 0059, 0106, 0169, 0181, and 0207. We report the absolute trajectory error of each sequence and the average result over all selected sequences. All experiments are conducted under an online RGB-D semantic SLAM setting. The system receives RGB-D images and semantic labels frame by frame, without using future-frame information. The first frame is used to initialize the semantic Gaussian map, and its camera coordinate system is used as the world reference frame. For each subsequent frame, the camera pose is first estimated during tracking, and the semantic Gaussian map is then updated during mapping. Final evaluation is performed every 5 frames.

B.2 Evaluation Metrics

We quantitatively evaluate Concord-SLAM from four aspects: pose estimation, geometric reconstruction, RGB rendering, and semantic mapping. For pose estimation, we report the absolute trajectory error root mean square error, denoted as ATE RMSE. The estimated trajectory is first aligned with the ground-truth trajectory, and the translational error is then computed in centimeters. Lower ATE RMSE indicates more accurate camera tracking. For geometric reconstruction, we report Depth L1 over valid depth pixels. Depth L1 measures the average absolute difference between rendered depth and observed or ground-truth depth. It is reported in centimeters, and lower values indicate more accurate geometry. For RGB rendering quality, we report PSNR, SSIM, and LPIPS. Higher PSNR and SSIM indicate better pixel-level and structural reconstruction quality, while lower LPIPS indicates better perceptual similarity. For semantic mapping, we use mean Intersection over Union, namely mIoU. It measures the overlap between predicted semantic regions and ground-truth semantic regions across categories. Higher mIoU indicates better semantic reconstruction. For runtime efficiency, we report Tracking FPS, Mapping FPS, and overall SLAM FPS. Tracking FPS measures the speed of pose estimation, Mapping FPS measures the speed of map update, and SLAM FPS reflects the overall online system throughput. Higher FPS indicates better runtime efficiency.

B.3 Baseline Methods

We compare Concord-SLAM with representative Gaussian SLAM and semantic Gaussian SLAM methods, including Gaussian-SLAM [Matsuki et al., 2024], GS-SLAM [Yan et al., 2024], SplaTAM [Keetha et al., 2024], SGS-SLAM [Li et al., 2024c], GS^{3}LAM [Li et al., 2024b], SemGauss-SLAM [Zhu et al., 2025b], and GSFF-SLAM [Lu et al., 2026]. Gaussian-SLAM, GS-SLAM, and SplaTAM are mainly used to evaluate localization, reconstruction, and RGB rendering performance. These methods adopt 3D Gaussian Splatting or related splat-based representations for dense RGB-D SLAM, and are representative baselines for Gaussian-map-based tracking and mapping. SGS-SLAM, GS^{3}LAM, SemGauss-SLAM, and GSFF-SLAM further introduce semantic representations or semantic optimization mechanisms into Gaussian SLAM. Therefore, they are used as the main semantic Gaussian SLAM baselines for semantic mapping comparison. For runtime efficiency, we compare Concord-SLAM with SGS-SLAM, GS^{3}LAM, and GSFF-SLAM using Tracking FPS, Mapping FPS, and overall SLAM FPS. For results already reported in the original papers, we preferentially adopt the published values. For missing scenes or missing metrics, we reproduce and supplement the results under the same experimental setting as the corresponding original method.

B.4 Implementation Settings

All experiments were conducted on a workstation equipped with a single NVIDIA GeForce RTX 5090 GPU with 32 GB of GDDR7 memory. Each sequence was processed on one GPU without multi-GPU parallelization. The runtime efficiency reported in Table 4 was measured under the same hardware setting. Unless otherwise specified, all baseline reproductions and ablation experiments were conducted using the same workstation to ensure a fair comparison. The scene is represented using isotropic 3D Gaussian primitives. Each Gaussian stores its position, color, opacity, scale, rotation, and a 16-dimensional semantic feature. The semantic feature is decoded into semantic logits through a lightweight convolutional semantic decoder. The decoder outputs predictions over 256 semantic categories. We use the Adam optimizer for camera tracking and map optimization. Different learning rates are used for different groups of parameters, including camera poses, Gaussian geometric parameters, color parameters, opacity, scale, rotation, and semantic features. This follows the common practice in Gaussian SLAM, where different Gaussian attributes have different numerical ranges and optimization sensitivities. On Replica, tracking is optimized for 40 iterations per frame. The first mapping frame is optimized for 1000 iterations, while each subsequent mapping frame is

Table 5: Key implementation hyperparameters of Concord-SLAM.

Hyperparameter	Replica	ScanNet
Geometry reliability temperature τ	8.0	8.0
RGB loss weight λ_I	0.5	0.5
Depth loss weight λ_D	1.0	1.0
Semantic loss weight λ_S	0.008	0.008
CRC soft semantic-consensus weight	0.015	0.015
Large-Gaussian regularization weight	0.02	0.02
Small-Gaussian regularization weight	0.002	0.002
Base opacity threshold τ_α	0.1	0.5
Depth residual multiplier κ	50	50
Boundary dilation radius r	1	1
Boundary opacity threshold τ_b	0.5	0.7
Tracking iterations	40	200
Mapping iterations per frame	60	60
First-frame mapping iterations	1000	1000
Keyframe interval	5	5
Configured mapping window size	24	10

Table 6: Architecture of the CRC module. Conv$k\times k(a, b)$ denotes a convolutional layer with kernel size $k\times k$, input channels a, and output channels b.

Component	Input	Layers	Output
Semantic encoder ϕ_s	$\hat{F}_t^s$, 16 ch	Conv3×3(16,64)+ReLU; Conv3×3(64,64)+ReLU	64 ch
Depth encoder ϕ_d	$\hat{D}_t$, 1 ch	Conv3×3(1,64)+ReLU; Conv3×3(64,64)+ReLU	64 ch
Normal encoder ϕ_n	$\hat{N}_t$, 3 ch	Conv3×3(3,64)+ReLU; Conv3×3(64,64)+ReLU	64 ch
Pairwise fusion	$[\phi_s, \phi_d]$, $[\phi_s, \phi_n]$, $[\phi_d, \phi_n]$	Conv3×3(128,64)+ReLU for each pair	Three 64-ch maps
Relation block H_θ	Concatenated pairwise features, 192 ch	Conv3×3(192,64)+ReLU; Conv3×3(64,64)+ReLU	64 ch
Semantic delta head	64 ch	Conv1×1(64,256)	ΔZ_t, 256 ch

optimized for 60 iterations. The keyframe interval is set to 5, and the mapping window size is set to 24. On ScanNet, due to stronger sensor noise, missing depth values, and more complex camera motion, tracking is optimized for 200 iterations per frame. The mapping iteration count and keyframe interval remain the same as those on Replica, while the mapping window size is set to 10. All sequences are processed with a frame stride of 1. Final evaluation is performed every 5 frames. Table 5 summarizes the key hyperparameters used in the final implementation. Unless otherwise specified, the loss weights are shared by Replica and ScanNet. The compact mapping objective is implemented with separate weights for RGB reconstruction, depth reconstruction, semantic supervision, cross-render semantic-geometric consistency, and Gaussian regularization.

B.5 Tracking Details

During tracking, Concord-SLAM mainly optimizes the camera pose of the current frame while keeping the semantic Gaussian map fixed. The pose of the current frame is initialized using a constant-velocity motion model. Given the initialized pose, the differentiable Gaussian renderer produces RGB, depth, opacity, and semantic predictions from the current Gaussian map. The tracking loss consists of RGB reconstruction loss, depth loss, and a weak semantic constraint. The RGB loss measures the difference between the rendered RGB image and the input RGB image. The depth loss measures the difference between rendered depth and observed depth over valid depth pixels. The weak semantic constraint provides additional semantic alignment between the rendered semantic prediction and the input semantic label. To improve tracking robustness, the loss is computed over high-confidence mapped regions. Specifically, regions with low rendered opacity are excluded because they are not sufficiently covered by the current Gaussian map. Pixels near semantic boundaries are also treated cautiously, since they are more likely to be affected by occlusion boundaries, depth discontinuities, and semantic-label misalignment. This design reduces the influence of unmapped regions and ambiguous boundary pixels on pose estimation. During tracking optimization, the system records the pose with the lowest tracking loss and uses it as the final estimated pose of the current frame.

B.6 Mapping Details

During mapping, Concord-SLAM optimizes the geometry, color, opacity, and semantic features of the Gaussian primitives under the estimated camera pose. The mapping objective includes RGB reconstruction, depth reconstruction, semantic supervision, Cross-Render Concordance, and Gaussian regularization. The RGB loss consists of an L1 reconstruction term and a structural similarity term. The depth loss is computed as the L1 error over valid depth regions. The semantic loss is implemented as pixel-wise cross-entropy between the decoded semantic logits and the semantic labels. To reduce forgetting and improve global map consistency, Concord-SLAM optimizes both the current frame and historical keyframes during mapping. At fixed iteration intervals, the current frame is used for optimization. For the remaining iterations, frames are randomly sampled from the keyframe set. This keyframe-replay strategy allows the map to absorb new observations while preserving consistency with previous views. The Cross-Render Concordance module is activated during mapping after the map becomes relatively stable. It fuses rendered semantic features, rendered depth, and normals derived from rendered depth, and predicts a correction term for semantic logits. For the CRC loss in Eq. (9), the normalization parameter is computed as

$$C_t = \max\left\{2,\ \min\left[C,\ 1 + \max_{p \in \Omega_{\text{img}}} Y_t^*(p)\right]\right\}. \tag{22}$$

Here, $C = 256$ is the number of semantic output channels, and $Y_t^*(p)$ is the observed semantic label ID, with labels indexed from zero. CRC is not used during tracking. During final evaluation, the corrected semantic output is used for semantic prediction, while the depth metric is computed from the depth rendered by the base Gaussian map, ensuring comparability with related methods. For semantic boundary regions, Concord-SLAM adopts Boundary-Native Splatting. This strategy inserts new Gaussians near semantic boundaries where opacity coverage is insufficient. By allocating additional primitives around under-covered semantic-boundary regions, BNS improves the representation of object boundaries and reduces cross-category semantic mixing.

B.7 Keyframe and Map Maintenance

Concord-SLAM saves keyframes at fixed frame intervals. Each keyframe stores the RGB image, depth map, semantic labels, and the estimated camera pose. During mapping, the system alternates between the current frame and randomly sampled keyframes to maintain temporal consistency across the reconstructed map. To control map size and reduce redundancy, Concord-SLAM periodically prunes abnormal Gaussian primitives. Gaussians with very low opacity or unstable scale values are removed. Gaussian scales are also regularized during optimization to avoid degenerate primitives. These maintenance operations help stabilize long-sequence mapping and improve runtime efficiency.

B.8 CRC Architecture

CRC is implemented as a lightweight convolutional multimodal coupling network. The network takes the rendered semantic feature map, rendered depth map, and depth-derived normal map as inputs. Three modality-specific encoders first map them into 64-channel embeddings. Pairwise semantic-depth, semantic-normal, and depth-normal features are then fused and passed to a relation block, which predicts a semantic logit residual ΔZ_t. No BatchNorm, LayerNorm, or GroupNorm is used. The activation function is ReLU. CRC is enabled during mapping but disabled during tracking. During evaluation, the CRC-corrected semantic logits are used for semantic prediction, while the rendered depth is kept from the base Gaussian map.

B.9 BNS Implementation

The base insertion mask consists of two conditions: low opacity and large missing-foreground depth residual. Specifically, the opacity condition is $\hat{A}_t < \tau_\alpha$, where $\tau_\alpha = 0.1$ on Replica and $\tau_\alpha = 0.5$ on ScanNet. The depth-residual condition is activated when $\hat{D}_t(p) > D_t^*(p)$ and $R_t^d(p) > 50 \cdot \text{median}(R_t^d)$. The semantic boundary map is computed from the semantic observation Y_t^* using 4-neighbor label differences, and both sides of a semantic transition are marked as boundary pixels. A max-pooling dilation with radius $r = 1$ is then applied. The BNS mask is defined as the dilated semantic boundary intersected with $\hat{A}_t < \tau_b$ and $D_t^* > 0$, where $\tau_b = 0.5$ on Replica and $\tau_b = 0.7$

on ScanNet. The final insertion mask is the union of the base mask and the BNS mask. Depth edges and normal edges are not directly used in the BNS mask; they contribute to CRC instead.

B.10 Depth-derived normals and geometric boundary response

Given a rendered depth map $\hat{D}_t$ and camera intrinsics $K = (f_x, f_y, c_x, c_y)$, we back-project each pixel into a 3D point map. Local tangent directions are computed by centered finite differences on the point map, and the normal is obtained by the cross product of the two tangent directions followed by ℓ_2 normalization with $\epsilon = 10^{-6}$. The depth edge response E_d is computed by first-order forward differences:

$$E_d(\hat{D}_t)(p) = \sqrt{\left(\nabla_x \hat{D}_t(p)\right)^2 + \left(\nabla_y \hat{D}_t(p)\right)^2 + \epsilon}, \tag{23}$$

where ∇_x and ∇_y denote horizontal and vertical forward differences with replicate padding, and $\epsilon = 10^{-8}$ is used for numerical stability. The normal variation response E_n measures local angular changes between adjacent unit normals:

$$E_n(\hat{N}_t)(p) = \frac{1}{2}\left(1 - |\langle \hat{n}_p, \hat{n}_{p+\mathbf{e}_x}\rangle| + 1 - \left|\langle \hat{n}_p, \hat{n}_{p+\mathbf{e}_y}\rangle\right|\right), \tag{24}$$

where $\hat{n}_p$ denotes the depth-derived normal at pixel p, and $\mathbf{e}_x$ and $\mathbf{e}_y$ denote horizontal and vertical neighboring offsets. The final geometric boundary response and reliability weight are

$$B_t^g = \frac{1}{2}\left(E_d(\hat{D}_t) + E_n(\hat{N}_t)\right), \qquad W_t^g = \exp(-8.0 B_t^g). \tag{25}$$

B.11 Gaussian Representation Analysis

We evaluate the effects of CRC and BNS on Replica-room1 using two groups of boundary-specific measurements: primitive separation and coverage, and effective support on both sides of semantic boundaries. We compare the full model with variants without CRC, without BNS, and without either module. In the following tables, "Neither" denotes the variant without both CRC and BNS. These measurements complement the overall semantic scores by examining the boundary effects targeted by the two modules.

Primitive separation and coverage. Boundary G fraction describes the proportion of Gaussian primitives allocated to boundary neighborhoods. CBGR and Spill measure cross-boundary mixing and contribution leakage, respectively, while Alpha coverage measures effective coverage. Table 7 shows that adding BNS to the CRC-equipped variant increases Alpha coverage from 96.67% to 99.79%, while reducing CBGR from 43.06% to 42.60% and Spill from 9.26% to 9.14%. The full model does not have the largest Boundary G fraction, but it has the lowest mixing and leakage and the highest coverage. Thus, the measured improvement concerns how effectively boundary primitives represent the scene, rather than their relative abundance alone.

Table 7: Gaussian separation and coverage in boundary neighborhoods on Replica-room1. All values are percentages. Boundary G fraction describes allocation; lower CBGR and Spill and higher Alpha coverage are better.

Variant	Boundary G fraction	CBGR ↓	Spill ↓	Alpha coverage ↑
Neither	28.21	45.88	11.26	95.66
w/o CRC	29.18	42.95	9.22	98.73
w/o BNS	29.61	43.06	9.26	96.67
Full	29.27	42.60	9.14	99.79

Effective support on both sides of boundaries. We evaluate primitive support within a 2-pixel neighborhood of semantic boundaries. BSR@2 measures the proportion of boundaries with effective support on both sides, whereas MSR@2 measures the proportion supported on only one side. In Table 8, adding BNS with CRC retained increases BSR@2 from 86.85% to 90.97% and reduces MSR@2 from 8.17% to 5.85%. The results directly support the intended role of BNS in strengthening bilateral boundary support. These mechanism-level observations are established on Replica-room1.

Table 8: Effective primitive support within 2 pixels of semantic boundaries on Replica-room1. All values are percentages.

Variant	BSR@2 ↑	MSR@2 ↓
Neither	85.65	9.28
w/o CRC	87.99	7.05
w/o BNS	86.85	8.17
Full	90.97	5.85

B.12 Semantic Mapping and Ablations on ScanNet

To examine semantic mapping under real RGB-D acquisition conditions, we evaluate SGS-SLAM, GS^{3}LAM, the full model, and the two single-module ablations on six ScanNet sequences: 0000, 0059, 0106, 0169, 0181, and 0207. We conduct five independent runs for each scene and report the average mIoU under a unified evaluation setting. Table 9 reports both the per-scene results and the average across the six scenes.

Table 9: Semantic mapping and module ablations on ScanNet, measured by mIoU (%). Each scene result is averaged over five independent runs. Higher is better. The two ablations are variants of Concord-SLAM.

Method	0000	0059	0106	0169	0181	0207	Avg.
SGS-SLAM	70.59	68.47	69.71	69.03	68.78	69.41	69.33
GS^{3}LAM	73.54	72.16	72.66	72.18	71.97	72.48	72.50
w/o BNS	73.88	73.24	73.04	74.10	73.84	73.73	73.64
w/o CRC	75.12	74.39	74.65	74.81	73.72	74.00	74.45
Full	75.84	75.05	75.12	74.65	74.43	75.20	75.05

The full model reaches an average mIoU of 75.05%, compared with 72.50% for GS^{3}LAM and 69.33% for SGS-SLAM. Its average also exceeds the two ablated variants, which obtain 73.64% without BNS and 74.45% without CRC. The advantage is an aggregate result: on sequence 0169, the variant without CRC obtains 74.81%, above the full model's 74.65%. Overall, the semantic-mapping advantage extends to these real indoor sequences, rather than being confined to the synthetic Replica benchmark.

B.13 Effects of CRC and BNS on Tracking and Reconstruction

Tracking and mapping share the same Gaussian representation, so changes in semantic boundary modeling may also affect pose estimation, depth reconstruction, and appearance rendering. We assess these effects through a four-way ablation on Replica-room1. Table 10 reports ATE RMSE, Depth L1, and PSNR.

Table 10: Tracking, depth, and rendering ablations on Replica-room1. ATE and Depth L1 are in centimeters, and PSNR is in dB.

Variant	ATE RMSE ↓	Depth L1 ↓	PSNR ↑
Neither	0.34	0.39	36.19
w/o CRC	0.28	0.34	36.20
w/o BNS	0.25	0.33	36.25
Full	0.22	0.32	36.33

The full model achieves the best values for all three reported metrics. Relative to the variant without either module, ATE decreases from 0.34 to 0.22 cm and Depth L1 decreases from 0.39 to 0.32 cm, while PSNR increases from 36.19 to 36.33 dB. Removing either module also degrades tracking and depth accuracy relative to the full model. These results show that the improvements in this scene are

not restricted to semantic outputs. The appearance gain is modest in absolute magnitude, and this single-scene analysis should be interpreted alongside the multi-scene results in the main paper.

B.14 Gaussian Count and Memory Overhead

We compare representation size and peak GPU memory usage on Replica-room1 under the same hardware and evaluation settings. Table 11 includes SGS-SLAM, GSFF-SLAM, GS^{3}LAM, Concord-SLAM without BNS, and the full model. Gaussian counts are recorded at 25%, 50%, and 75% sequence progress, together with the final count, the peak count, and peak VRAM. Counts are expressed in millions (M), and memory is expressed in GiB.

Table 11: Gaussian counts and peak VRAM on Replica-room1. The five Gaussian-count columns are in millions; lower values indicate smaller representations or lower memory usage. w/o BNS denotes Concord-SLAM without BNS.

Method	25% G	50% G	75% G	Final G	Peak G	Peak VRAM (GiB)
SGS-SLAM	2.10	2.17	2.26	2.15	2.32	11.18
GSFF-SLAM	2.13	2.19	2.31	2.23	2.36	10.74
GS^{3}LAM	3.10	3.16	3.27	3.21	3.32	9.65
w/o BNS	3.77	3.82	3.95	3.85	3.98	12.17
Full	5.12	5.20	5.31	5.21	5.36	14.58

BNS introduces a measurable resource cost. Compared with the variant without BNS, the full model uses approximately 35% more Gaussians at the sampled stages, while peak VRAM increases by 19.8%, from 12.17 to 14.58 GiB. It also uses more Gaussians and memory than the external baselines. This cost accompanies the improved boundary coverage and bilateral support measured in Appendix B.11.

Between 25% and 75% sequence progress, the Gaussian-count increase remains below 4.8% for both Concord-SLAM variants, and each final count is lower than its corresponding peak. These observations are consistent with coverage checking and periodic pruning limiting redundant growth in the evaluated sequence. They do not establish a bound for arbitrarily long trajectories or large-scale environments; the additional representation and memory requirements remain a limitation of BNS.

B.15 Robustness to Noisy Semantic Inputs

We evaluate controlled semantic-input perturbations on ScanNet-scene0059_00 using the full model and the variants without BNS or CRC. This scene contains adjacent walls, floors, tables, chairs, and small objects, providing varied semantic boundaries and local occlusions. Spatial noise randomly replaces labels at 1%, 3%, 5%, or 10% of valid semantic pixels. Temporal flicker shifts semantic masks by up to 5 pixels in 10%, 20%, or 30% of frames. For each noise setting, we perform five independent runs and provide identical noisy inputs to all variants within each run. Results are reported as mean $\pm$ standard deviation over these five independent runs.

Table 12 distinguishes nominal noise intensity from the fraction of pixels whose labels actually change. The boundary change rate is the relative change in the number of four-neighbor semantic boundaries. In particular, temporal intensity specifies the fraction of perturbed frames, not the fraction of altered semantic pixels. The slightly negative boundary change rates for temporal flicker arise from image-edge cropping when masks are shifted.

Tables 13 and 14 show that the full model retains the best semantic, boundary, and tracking results among the three variants at every tested noise level. Nevertheless, input noise reduces accuracy: under 10% spatial noise, its mIoU decreases from 75.05% to 73.14% and ATE increases from 7.53 to 8.36 cm. The input boundary count increases by 3641.15%, while its final and peak Gaussian counts increase by 5.88% and 8.22%, respectively. Thus, the large increase in spurious input boundaries is not accompanied by proportional representation growth.

Under 30% temporal flicker, the full model obtains 71.65% mIoU and 8.50 cm ATE, while its final and peak Gaussian counts increase by 3.56% and 5.29% relative to clean input. Its Gaussian counts also remain below those of the variant without CRC. These observations are consistent with the combined

Table 12: Semantic-input perturbations on ScanNet-scene0059_00. All numeric columns are percentages. S denotes spatial noise and T denotes temporal flicker; these identifiers are used in the following result tables.

Setting	Noise type	Intensity	Changed pixels	Boundary change rate
Clean	None	0.00	0.00	0
S1	Spatial	1.00	1.00	486.61
S3	Spatial	3.00	3.00	1054.54
S5	Spatial	5.00	5.00	1863.69
S10	Spatial	10.00	10.00	3641.15
T10	Temporal	10.00	0.20	-0.0243
T20	Temporal	20.00	0.40	-0.0547
T30	Temporal	30.00	0.60	-0.0933

Table 13: Semantic, boundary, and tracking performance under the perturbations in Table 12. mIoU and Boundary F1@2 are percentages; ATE RMSE is in centimeters. Values are mean $\pm$ standard deviation over five independent runs.

Setting	Variant	mIoU ↑	Boundary F1@2 ↑	ATE RMSE ↓
Clean	Full	75.05 ± 0.18	80.15 ± 0.25	7.53 ± 0.15
Clean	w/o BNS	73.24 ± 0.24	78.75 ± 0.31	7.72 ± 0.18
Clean	w/o CRC	74.39 ± 0.22	78.20 ± 0.34	7.67 ± 0.17
S1	Full	74.94 ± 0.20	79.54 ± 0.28	7.65 ± 0.17
S1	w/o BNS	73.08 ± 0.27	78.10 ± 0.35	7.83 ± 0.20
S1	w/o CRC	74.05 ± 0.25	77.25 ± 0.39	7.84 ± 0.20
S3	Full	74.57 ± 0.24	79.12 ± 0.32	7.91 ± 0.20
S3	w/o BNS	72.61 ± 0.31	77.18 ± 0.42	8.12 ± 0.23
S3	w/o CRC	73.38 ± 0.30	75.92 ± 0.47	8.22 ± 0.24
S5	Full	73.45 ± 0.29	77.40 ± 0.40	8.24 ± 0.24
S5	w/o BNS	71.18 ± 0.38	74.86 ± 0.51	8.46 ± 0.28
S5	w/o CRC	71.82 ± 0.36	73.08 ± 0.56	8.62 ± 0.29
S10	Full	73.14 ± 0.34	76.64 ± 0.48	8.36 ± 0.28
S10	w/o BNS	69.72 ± 0.46	71.98 ± 0.65	8.72 ± 0.33
S10	w/o CRC	69.98 ± 0.44	69.74 ± 0.72	8.91 ± 0.35
T10	Full	72.75 ± 0.26	76.64 ± 0.37	7.90 ± 0.21
T10	w/o BNS	70.96 ± 0.35	73.72 ± 0.48	8.12 ± 0.25
T10	w/o CRC	71.18 ± 0.34	72.86 ± 0.52	8.15 ± 0.25
T20	Full	72.14 ± 0.31	75.90 ± 0.44	8.16 ± 0.25
T20	w/o BNS	69.54 ± 0.41	71.64 ± 0.58	8.40 ± 0.29
T20	w/o CRC	69.78 ± 0.40	70.48 ± 0.64	8.48 ± 0.30
T30	Full	71.65 ± 0.38	75.06 ± 0.52	8.50 ± 0.30
T30	w/o BNS	68.03 ± 0.49	69.42 ± 0.68	8.82 ± 0.35
T30	w/o CRC	68.25 ± 0.48	67.76 ± 0.76	8.91 ± 0.36

Table 14: Final and peak Gaussian counts under the perturbations in Table 12. Counts are in millions and reported as mean ± standard deviation over five independent runs. Lower counts indicate smaller representations and should be considered together with the accuracy results.

Setting	Variant	Final G (M)	Peak G (M)
Clean	Full	6.46 ± 0.04	6.81 ± 0.06
Clean	w/o BNS	4.72 ± 0.06	5.02 ± 0.03
Clean	w/o CRC	6.52 ± 0.06	6.90 ± 0.07
S1	Full	6.55 ± 0.06	6.85 ± 0.06
S1	w/o BNS	4.78 ± 0.06	5.08 ± 0.04
S1	w/o CRC	6.65 ± 0.07	7.01 ± 0.08
S3	Full	6.47 ± 0.07	6.90 ± 0.07
S3	w/o BNS	4.82 ± 0.04	5.15 ± 0.08
S3	w/o CRC	6.82 ± 0.08	7.20 ± 0.02
S5	Full	6.70 ± 0.08	7.26 ± 0.09
S5	w/o BNS	4.91 ± 0.05	5.26 ± 0.09
S5	w/o CRC	7.10 ± 0.09	7.51 ± 0.10
S10	Full	6.84 ± 0.09	7.37 ± 0.10
S10	w/o BNS	5.02 ± 0.09	5.39 ± 0.09
S10	w/o CRC	7.49 ± 0.11	7.93 ± 0.12
T10	Full	6.50 ± 0.07	7.03 ± 0.08
T10	w/o BNS	4.79 ± 0.05	5.12 ± 0.08
T10	w/o CRC	6.63 ± 0.08	7.12 ± 0.08
T20	Full	6.62 ± 0.08	7.11 ± 0.09
T20	w/o BNS	4.86 ± 0.07	5.23 ± 0.09
T20	w/o CRC	6.79 ± 0.09	7.28 ± 0.10
T30	Full	6.69 ± 0.09	7.17 ± 0.07
T30	w/o BNS	4.94 ± 0.09	5.31 ± 0.10
T30	w/o CRC	6.98 ± 0.10	7.49 ± 0.12

effect of coverage checking, CRC optimization, and periodic pruning, although the experiment does not isolate the contribution of each maintenance operation. The results characterize robustness to these controlled perturbations on one real indoor sequence; they do not imply immunity to arbitrary frontend errors or dynamic scenes.

B.16 Tracking and Rendering Evaluation on TUM RGB-D

We further evaluate tracking and rendering on five TUM RGB-D sequences: fr1/desk1, fr1/desk2, fr1/room, fr2/xyz, and fr3/office. We perform five independent runs for each scene and report the average results across the five sequences. Camera tracking is evaluated against the provided reference trajectories. This evaluation reports ATE RMSE, PSNR, SSIM, and LPIPS; it does not report dense semantic accuracy because unified dense semantic annotations are not available for this evaluation.

Table 15: Tracking and rendering results averaged over five TUM RGB-D sequences, with five independent runs per scene. ATE RMSE is measured in centimeters and PSNR in dB.

Method	ATE RMSE ↓	PSNR ↑	SSIM ↑	LPIPS ↓
SGS-SLAM	5.08	15.54	0.58	0.84
GSFF-SLAM	3.95	20.57	0.74	0.31
GS^{3}LAM	4.87	19.63	0.75	0.49
Ours	3.75	21.66	0.81	0.31

As shown in Table 15, Concord-SLAM achieves the lowest average ATE RMSE and the highest PSNR and SSIM among the compared methods, while tying with GSFF-SLAM for the lowest LPIPS. These

results complement the Replica and ScanNet evaluations by demonstrating competitive tracking and rendering under another set of real indoor RGB-D acquisition conditions. The evaluation broadens the indoor benchmark coverage without extending the demonstrated scope to outdoor, dynamic, or open-vocabulary settings.

C Limitations

Although Concord-SLAM improves semantic Gaussian SLAM through cross-render consistency and boundary-native Gaussian growth, several limitations remain. First, Concord-SLAM is mainly designed for static indoor scenes. Dynamic objects may introduce inconsistent observations across frames and affect both Gaussian optimization and semantic consistency modeling. Handling dynamic scenes remains an important direction for future work. Second, BNS introduces additional Gaussians around under-covered semantic-boundary regions. Although this improves boundary representation, it may increase memory usage and optimization cost in large-scale scenes. More efficient boundary-adaptive Gaussian management strategies are needed to balance mapping quality and computational efficiency. Finally, CRC assumes that semantic, depth, and normal renderings should be structurally consistent. This assumption is generally valid in static scenes, but errors in depth estimation, noisy normals, or inaccurate semantic predictions may still affect the reliability of the consistency signals. Improving the robustness of cross-render consistency under noisy observations is another promising direction.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The main claims are stated in the abstract and introduction, and are supported by the experimental results.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: The paper discusses limitations related to generalization, dependence on input quality, and computational cost.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

Answer: [N/A]

Justification: The paper does not present theoretical results or formal proofs.

Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: The paper reports datasets, evaluation metrics, baselines, implementation details, and experimental settings needed to reproduce the main results.

   Guidelines:

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

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

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

Answer: [No]

Justification: The code was shared with reviewers during peer review but is not publicly accessible. Appendix B describes the datasets, implementation settings, and evaluation protocol.

Guidelines:

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

6. **Experimental setting/details**

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

Answer: [Yes]

Justification: The paper specifies the experimental settings, including datasets, evaluation protocol, training details, hyperparameters, optimization settings, and baseline configurations.

Guidelines:

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

7. **Experiment statistical significance**

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

Answer: [Yes]

Justification: The paper reports results over multiple runs and provides statistical information such as mean and standard deviation for the main experiments. These statistics characterize the variability caused by repeated runs under the same experimental protocol.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

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

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: The paper reports the GPU type and memory used for all experiments, and reports runtime efficiency in terms of Tracking FPS, Mapping FPS, and overall SLAM FPS.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: The research follows the NeurIPS Code of Ethics and uses standard public datasets without collecting private user data or involving human-subject experiments.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: The paper discusses potential positive impacts for robotic perception, 3D scene understanding, and embodied AI, as well as possible risks related to privacy-sensitive deployment and surveillance.

Guidelines:

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

11. **Safeguards**

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

    Answer: [N/A]

    Justification: The paper does not release high-risk generative models, scraped datasets, or assets with substantial misuse risk.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: The paper properly credits the existing datasets, codebases, and baseline methods used in the experiments, and follows their licenses and terms of use.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.

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

13. **New assets**

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

    Answer: [No]

    Justification: The camera-ready submission includes implementation details in Appendix B and does not include a separate code package.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing or human-subject experiments.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: The paper does not involve human-subject research and therefore does not require IRB approval.

    Guidelines:

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

16. **Declaration of LLM usage**

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

    Answer: [N/A]

    Justification: LLMs are not used as an important, original, or non-standard component of the core research method.

    Guidelines:

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