LDO-SLAM: Hierarchical Semantic-Enhanced 3DGS-SLAM with Robust Loop Closure

1st Jichao Jiao
Beijing University of Posts and Telecommunications
Beijing, China
jiaojichao@gmail.com

2nd Yingchao Zeng
Beijing University of Posts and Telecommunications
Beijing, China
yingchaoz@bupt.edu.cn

***Abstract*—Recent advances in 3D Gaussian Splatting (3DGS) have improved scene representation efficiency and fidelity, but existing 3DGS-SLAM systems suffer from tracking drift, limited global optimization, and limited online semantic reasoning. We present LDO-SLAM, a dense RGB-D SLAM framework that integrates 3DGS with hierarchical semantic inference and loop closure. Our system fuses closed-set object detection and open-vocabulary vision-language cues into 3D Gaussians, enabling semantic submap construction and a two-stage, semantic-guided loop closure with global graph optimization. This design improves robustness to perceptual aliasing while keeping the pipeline lightweight. Experiments on Replica, TUM-RGBD, and ScanNet show competitive localization and mapping quality, while maintaining an end-to-end throughput of ≈2 FPS on a single RTX 3090 GPU.**

***Index Terms*—3D Gaussian Splatting, Visual SLAM, Semantic Enhancement, Loop Closure**

I. Introduction

Visual SLAM is fundamental for motion estimation and scene understanding [1]. While classical systems like ORB-SLAM enable diverse applications, recent efforts focus on advancing scene representation for better fidelity and efficiency.

Conventional mesh-based representations suffer from high memory demands and limited scalability. Neural Radiance Fields (NeRF) [2] offer continuous high-fidelity alternatives, but intensive per-ray rendering impedes real-time deployment. 3D Gaussian Splatting (3DGS) [3] provides a compelling compromise, modeling scenes as Gaussian primitives with differentiable attributes, achieving NeRF-comparable quality with superior efficiency.

However, current 3DGS-SLAM methods lack global pose optimization, causing drift and map inconsistencies. While NeRF-based (Photo-SLAM [4]) and implicit methods (Go-SLAM [5], Loopy-SLAM [6]) explore loop closure, they rely on inefficient feature-matching or require extensive retraining, limiting applicability. Moreover, integrating semantic information with 3DGS for joint localization and mapping remains insufficiently addressed under interactive-rate constraints, as current methods require pre-segmented data, violating SLAM's online requirements.

We propose an interactive-rate, semantically aware dense SLAM framework based on 3DGS (Fig. 1), comprising semantic feature extraction (bounding-box detections from YOLOv5, not masks), 3D Gaussian mapping, and global optimization jointly exploiting semantic and geometric cues. Our contributions:

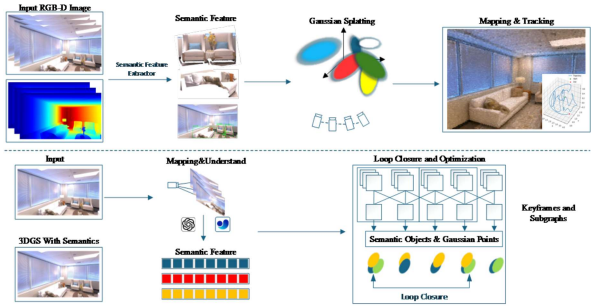

Fig. 1. Overview of the proposed semantic 3D Gaussian Splatting SLAM framework. Semantic and geometric data are combined for efficient mapping and robust camera tracking.

- We introduce a semantic-enhanced 3DGS representation and a visibility-aware association strategy to propagate 2D semantic cues to 3D Gaussians in an online SLAM setting.
- We integrate semantic cues into the pipeline with semantic submap descriptors, two-stage loop closure, and factor-graph optimization with semantic landmarks to improve global consistency.
- Experiments on synthetic and real-world RGB-D benchmarks show improved localization and mapping quality compared with recent 3DGS-SLAM baselines, while operating at ≈2 FPS end-to-end on a single GPU.

II. Related Works

A. Visual SLAM

Visual SLAM evolved from sparse feature-based frameworks (ORB-SLAM [1]) offering robust tracking but limited dense reconstruction and no semantic awareness. Dense methods improve fidelity via volumetric mapping, and learning-based pipelines (DROID-SLAM [7]) leverage neural architectures. However, they suffer from memory demands, drift, and scalability issues.

B. 3DGS SLAM

3D Gaussian Splatting (3DGS [3]) enables efficient, high-fidelity scene representation. Recent systems (SplaTAM [8],

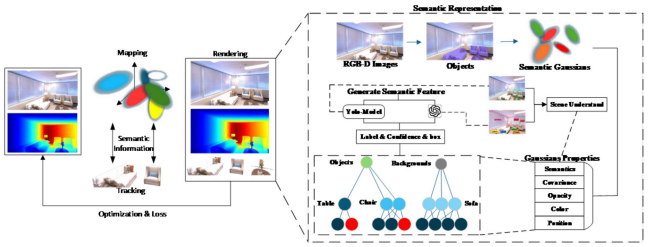

Fig. 2. Overview of the proposed semantic-enhanced 3D Gaussian SLAM pipeline.

GS-SLAM [9]) validate 3DGS for SLAM but face limitations: (1) lack of global optimization, using only local alignment; (2) inefficient loop closure with large keyframe storage (Go-SLAM [5]); (3) minimal semantic integration.

LoopSplat [10] introduces 3DGS-based loop closure via direct 3DGS registration, achieving globally consistent reconstruction on RGB-D sequences. However, it operates purely on geometric and visual features without semantic reasoning, limiting robustness in perceptually aliasing environments. RGBDS-SLAM [11] employs multi-level pyramid Gaussian splatting with tightly-coupled RGB-depth-semantic optimization, achieving high-fidelity reconstruction (PSNR 38.85, mIoU 94.32% on Replica). However, it relies on ground-truth semantic labels for training and lacks open-vocabulary capability or object-level loop closure mechanisms. Our method addresses these gaps through open-vocabulary semantic guidance, lightweight loop verification (85ms), and factor graph optimization with semantic landmarks, balancing efficiency with semantic reasoning.

C. Semantic Feature SLAM

Traditional SLAM rarely considers scene semantics, limiting robotic reasoning. Semantic-NeRF and neural approaches (iMAP [12], NICE-SLAM [13]) improve object recognition but increase cost. Semantic 3DGS systems (SemGauss-SLAM [14], SGS-SLAM [15], OpenGS-SLAM [16]) enable dense semantic mapping but depend on offline pre-segmentation.

Open-vocabulary SLAM (OVO-SLAM [17], SNI-SLAM [18]) leverages vision-language models for broader coverage. However, OVO-SLAM struggles with computational overhead, while SNI-SLAM faces consistency challenges. Registration-first approaches (ProFuse, Dr.Splat) achieve superior accuracy via offline fusion but sacrifice online adaptability. Our method bridges this gap through asynchronous VLM invocation and Bayesian fusion, balancing efficiency with open-vocabulary capability. Semantic loop closure (e.g., SemGauss-SLAM) improves consistency but lacks explicit object-level geometric verification, which our two-stage approach addresses.

III. Method

We propose a semantic-enhanced 3D Gaussian Splatting (3DGS) SLAM framework that fuses geometric and semantic cues in a unified scene representation (Fig. 2). The pipeline integrates closed-set object detection and open-set vision-language reasoning through a multi-stage semantic fusion, and embeds semantic labels into 3D Gaussian primitives to enable robust mapping, localization, and data association. Semantic-guided loop closure and pose graph optimization further improve global consistency and trajectory accuracy.

A. Semantic-Enhanced Gaussian Representation

We represent the scene as anisotropic 3D Gaussians with semantic attributes:

$$\mathcal{G}_i = (\mu_i, \Sigma_i, \mathbf{c}_i, o_i, \mathbf{p}_i)$$

where $\mu_i \in \mathbb{R}^3$ is the Gaussian mean (3D position), $\Sigma_i \in \mathbb{R}^{3\times3}$ is the covariance matrix, $\mathbf{c}_i$ is the color, o_i is the opacity, and $\mathbf{p}_i = [p_{i,1}, \ldots, p_{i,C}]$ is a semantic distribution over C categories.

Visibility-aware Semantic Assignment: Given camera pose (R, t) and intrinsics K, we project each Gaussian center and approximate its footprint as an ellipse via projected covariance $\Sigma_p = J(R\Sigma_i R^\top)J^\top$, where J is the projection Jacobian. We convert the ellipse to an axis-aligned bounding box (AABB) using a 2-sigma contour. Occlusion is handled via a depth test on the depth map at the projected center: if the Gaussian depth is behind the observed depth by more than $\delta_z = 0.1$m, we skip semantic update. For each detected box b_j, we compute IoU with the Gaussian AABB and update semantics when IoU$> \theta_{\text{IoU}}$ (0.3).

Semantic Distribution Update: Each Gaussian maintains a semantic distribution updated via exponential moving average:

$$\mathbf{p}_i^{(t+1)} = \lambda \mathbf{p}_i^{(t)} + (1-\lambda)\mathbf{p}_i^{\text{obs}}$$

where $\lambda = 0.9$ controls temporal smoothing and $\mathbf{p}_i^{\text{obs}}$ is the observed distribution from detection. Semantic entropy $H_i = -\sum_c p_{i,c} \log p_{i,c}$ quantifies uncertainty, guiding adaptive label assignment and loop closure verification. Unmatched Gaussians increment a miss counter and revert to background after $N_{miss} = 5$ frames.

Semantic-Geometric Consistency: We enforce consistency between semantic labels and geometric structure through a regularization term:

$$\mathcal{L}_{\text{consistency}} = \sum_{i,j\in\mathcal{N}} w_{ij} \cdot \mathbb{1}[s_i \neq s_j] \cdot \exp\left(-\|\mu_i - \mu_j\|^2/\sigma_{\text{cons}}^2\right)$$

where w_{ij} penalizes semantic discontinuities between spatially close Gaussians ($\sigma_{\text{cons}} = 0.05$m), encouraging smooth semantic boundaries aligned with geometric edges.

B. Hierarchical Semantic Feature Fusion

Motivation: Closed-set detectors (YOLOv5) provide high-confidence predictions for common objects but fail on novel categories, while open-vocabulary vision-language models (VLMs) offer broader coverage but with higher uncertainty. We adopt a calibrated confidence fusion rule that combines both sources while keeping VLM calls asynchronous.

Fusion Rule (Calibrated): Given a detector prediction with label $\hat{s}_{\text{YOLO}}$ and confidence $\text{Conf}_{\text{YOLO}} \in [0,1]$, and an

optional VLM prediction with label $\hat{s}_{\text{VLM}}$ and confidence $\text{Conf}_{\text{VLM}} \in [0,1]$, we compute reliability scores

$$r_{\text{YOLO}} = \text{IoU}(b_{\text{YOLO}}, \text{AABB}(\mathcal{G}_i)) \cdot \text{Conf}_{\text{YOLO}},$$
$$r_{\text{VLM}} = \text{Conf}_{\text{VLM}}.$$

then set the fusion weight

$$\alpha = \frac{r_{\text{YOLO}}^{\beta}}{r_{\text{YOLO}}^{\beta} + r_{\text{VLM}}^{\beta} + \epsilon}.$$

where $\beta = 2.0$ and $\epsilon = 10^{-6}$. The fused semantic confidence is

$$\text{SemConf}_i = \alpha \, \text{Conf}_{\text{YOLO}} + (1-\alpha) \, \text{Conf}_{\text{VLM}}.$$

If the VLM is not invoked, we set $\text{Conf}_{\text{VLM}} = 0$ (equivalently $\alpha = 1$), so the update reduces to detector-only. We adopt the label with maximum fused confidence, and encode the selected label as $\mathbf{p}_i^{\text{obs}}$ for the EMA update described above. VLM inference is triggered only when $\text{Conf}_{\text{YOLO}} < 0.6$ or when the detector fails to provide a reliable label.

Label Space Unification: YOLO predicts over a fixed set $\mathcal{C}_{\text{closed}}$ of $C = 90$ COCO classes, while the VLM can propose arbitrary text labels. We maintain a global vocabulary $\mathcal{C}_{\text{all}}$ (label-to-index map) initialized with COCO classes. For efficiency, we cap $|\mathcal{C}_{\text{all}}|$ by $C_{\max}$ and store per-Gaussian semantics sparsely (top-L labels); labels beyond the cap are mapped to an `other` token to avoid frequent dense reallocation (we use $C_{\max} = 256$ and $L = 5$).

C. Semantic-Guided Loop Closure and Optimization

Subgraph Construction: We group temporally adjacent keyframes into subgraphs of size $M \in [3,5]$. Each keyframe stores a semantic histogram over detected labels; a subgraph descriptor aggregates its keyframes using a TF-IDF weighted bag-of-words (BoW) vector $\mathbf{V}^{\text{BoW}}$.

Two-Stage Loop Retrieval and Verification: Unlike traditional BoW-based loop closure that relies solely on visual features, our method jointly exploits semantic and geometric cues to reduce false positives in perceptually aliasing environments (e.g., repetitive corridors, similar rooms).

Coarse Stage: For a query subgraph i, we retrieve candidates by computing semantic similarity with a temporal exclusion window:

$$S_{\text{coarse}}(i,j) = \cos(\mathbf{V}_i^{\text{BoW}}, \mathbf{V}_j^{\text{BoW}}) \cdot \mathbb{1}[|t_i - t_j| > \Delta t_{\min}]$$

where t_i is the keyframe index (or timestamp) of subgraph i, and $\Delta t_{\min} = 100$ (in frames) suppresses near-duplicate matches. We keep top-$K = 5$ matches passing threshold $\theta_{\text{loop}} = 0.75$.

Fine Stage: Each candidate undergoes verification using: (1) **Object-level geometric consistency**: We match semantic landmarks between subgraphs and compute relative pose via RANSAC on landmark correspondences. (2) **Semantic consistency check**: We verify that semantic distributions of matched landmarks have low KL-divergence (< 0.3). This hierarchical semantic-geometric fusion significantly reduces false positives compared to purely geometric or visual methods.

Factor Graph Optimization with Semantic Landmarks: We build a factor graph $G = (V, E)$ with pose nodes $\mathbf{x}_{1:T}$ and semantic landmark nodes $\mathbf{l}_{1:K}$.

Landmark Instantiation and Parameterization: Semantic landmarks are object-level representations instantiated when a detected object (YOLO box) persists across ≥ 3 keyframes with consistent label (KL-divergence < 0.2 between semantic distributions). Each landmark $\mathbf{l}_k$ is parameterized as $(\mu_k, \Sigma_k, \mathbf{p}_k)$, where $\mu_k \in \mathbb{R}^3$ is the 3D centroid (mean of associated Gaussian centers), $\Sigma_k \in \mathbb{R}^{3\times3}$ is the spatial covariance, and $\mathbf{p}_k = [p_{k,1}, \ldots, p_{k,C}]$ is the semantic distribution (aggregated from Gaussians via weighted averaging).

Observation Model and Data Association: At each keyframe, we project existing landmarks into the image and associate them with detected boxes via IoU matching (threshold 0.4). The observation factor penalizes reprojection error and semantic inconsistency:

$$f_{\text{obs}}(\mathbf{x}_t, \mathbf{l}_k) = \|\pi(\mathbf{x}_t, \mu_k) - \mathbf{z}_t^k\|_{\Sigma_{\text{obs}}}^2 + \lambda_{\text{sem}} \cdot \text{KL}(\mathbf{p}_k \| \mathbf{p}_t^k)$$

where $\pi(\cdot)$ is the projection function, $\mathbf{z}_t^k$ is the observed box center, and $\mathbf{p}_t^k$ is the observed semantic distribution. Unmatched landmarks increment a miss counter; after 10 consecutive misses, they are marginalized out.

Semantic Similarity Factors: Between co-visible landmarks $\mathbf{l}_i, \mathbf{l}_j$ in the same keyframe, we add semantic similarity factors based on label distribution consistency:

$$f_{\text{sem}}(\mathbf{l}_i, \mathbf{l}_j) = w_{\text{sem}} \cdot \text{KL}(\mathbf{p}_i \| \mathbf{p}_j) \cdot \mathbb{1}[\arg\max \mathbf{p}_i = \arg\max \mathbf{p}_j]$$

where $w_{\text{sem}} = 0.1$ weights the semantic constraint. This encourages consistent semantic distributions for landmarks with the same predicted class.

Factor Graph Edges: The graph includes four edge types: (1) odometry factors (consecutive poses $\mathbf{x}_t, \mathbf{x}_{t+1}$), (2) loop-closure factors (verified loop pairs), (3) landmark observation factors (pose-landmark pairs), and (4) semantic similarity factors (landmark-landmark pairs). Each landmark maintains entropy $H_k = -\sum_c p_{k,c} \log p_{k,c}$; high-entropy landmarks ($H_k > 0.5$) are down-weighted by factor $\exp(-2H_k)$ and marginalized earlier. We adopt incremental optimization (iSAM2) with periodic marginalization (every 10 keyframes) for scalability.

IV. Experiment

A. Datasets and Implementation

We evaluate on **Replica** [22] (synthetic indoor), **TUM RGB-D** [23] (real-world with noise), and **ScanNet** [24] (large-scale with semantics). Experiments run on NVIDIA RTX 3090. Table VI lists key hyperparameters. *Fairness statement:* For all baselines, we follow the evaluation protocol and settings reported in the original papers; when official code is available we use it, otherwise we report the published numbers.

TABLE I
LOCALIZATION PERFORMANCE ATE RMSE (CM) ON THE REPLICA DATASET. BOLD NUMBERS DENOTE THE BEST RESULTS.

Method	rm1	rm2	off2	off3	**Avg.**
NICE-SLAM [13]	1.31	1.07	1.06	1.10	1.06
ESLAM [19]	0.70	0.52	0.58	0.72	0.63
Point-SLAM [20]	0.41	0.41	0.54	0.69	0.52
GO-SLAM [5]	0.29	0.29	0.39	0.39	0.35
SplaTAM [8]	0.40	0.29	0.29	0.32	0.38
MonoGS [9]	0.43	0.31	0.31	0.31	0.79
Photo-SLAM [4]	0.39	0.52	0.78	0.58	0.67
SNI-SLAM [18]	0.55	0.45	0.33	0.62	0.46
SemGauss-SLAM [14]	0.42	0.27	0.32	0.36	0.33
LDO-SLAM (Ours)	**0.19**	**0.22**	**0.22**	**0.12**	**0.25**

TABLE II
LOCALIZATION PERFORMANCE ATE RMSE (CM) ON TUM-RGBD. BOLD NUMBERS DENOTE THE BEST RESULTS.

Method	fr1/desk	fr2/xyz	fr3/off	**Avg.**
NICE-SLAM [13]	4.26	6.19	3.87	4.77
ESLAM [19]	2.47	1.11	2.42	2.00
Point-SLAM [20]	4.34	1.31	3.48	3.04
SplaTAM [8]	3.35	1.24	5.16	3.25
Gaussian-SLAM [21]	2.73	1.39	5.31	3.14
Photo-SLAM [4]	2.60	**0.35**	**1.00**	**1.32**
LoopSplat [10]	2.08	1.58	3.22	1.82
LDO-SLAM (Ours)	**1.28**	1.32	1.43	**1.32**

Parameter Selection: $\theta_{\text{IoU}} = 0.3$ balances precision/recall in semantic assignment; $\beta = 2.0$ controls fusion sharpness between YOLO and VLM; $\theta_{\text{loop}} = 0.75$ is tuned to achieve high loop precision in our evaluation ($> 95\%$ on Replica/room0); $\lambda_{\text{sem}} = 0.5$ balances geometric and semantic factors in optimization. Other implementation details (EMA factor $\lambda = 0.9$, depth tolerance 0.1m, subgraph size 3–5 keyframes) follow standard practices.

Evaluation Metrics: We report ATE RMSE (cm) for tracking accuracy and PSNR/SSIM/LPIPS for reconstruction quality. For semantic evaluation, since 3DGS lacks standard 3D semantic protocols, we compute 2D projected per-pixel semantic accuracy and mIoU on Replica/ScanNet test views. Specifically, we render a semantic label map with the same splatting renderer using depth ordering (z-buffer) and alpha compositing, assign each pixel the label of the Gaussian with the highest accumulated contribution, and ignore void/unknown pixels in the ground truth. We compute mIoU over the dataset-provided semantic label set; open-vocabulary predictions are mapped to the evaluation label space via a simple synonym/keyword table, and unmapped labels are treated as unknown. Rendering and ground-truth are aligned on the same pixel grid (same intrinsics and resolution for each view).

B. Experimental Results

1) Camera Tracking: LDO-SLAM significantly improves localization over NICE-SLAM [13], ESLAM [19], SplaTAM [8], and MonoGS [9], consistently achieving lower trajectory errors across synthetic and real-world datasets.

TABLE III
LOCALIZATION PERFORMANCE ATE RMSE (CM) ON SCANNET DATASET. BOLD NUMBERS DENOTE THE BEST RESULTS.

Method	0059	0181	0207	**Avg.**
NICE-SLAM [13]	14.0	13.4	**6.2**	10.7
Point-SLAM [20]	7.8	14.8	9.5	12.2
MonoGS [9]	32.1	21.8	7.9	15.2
SplaTAM [8]	10.1	11.1	7.5	11.9
Gaussian-SLAM [21]	12.8	21.0	14.3	17.1
SemGauss-SLAM [14]	7.97	9.78	8.97	-
LDO-SLAM (Ours)	**7.6**	**8.5**	6.6	**8.7**

2) Rendering Quality: Compared to Point-SLAM [20], Loopy-SLAM [6], and SplaTAM [8], our approach generates reconstructions with higher perceptual quality and photometric consistency.

Fig. 3. Semantic object detection in reconstructed scenes. Top: Original RGB; middle: Detected objects (YOLO/VLM) with bounding boxes in 3D; bottom: Highlighted semantic objects.

We incorporate interactive-rate semantic object extraction using YOLOv5 [25] for bounding-box detection (not masks), enabling efficient identification of objects (sofas, chairs, tables) in reconstructed 3D environments (Fig. 3). This supports semantic navigation and robotic manipulation.

3) Semantic Segmentation Quality: We project Gaussians to 2D and compute per-pixel semantic labels (using depth ordering and pixel-wise maximum contribution; void pixels are ignored). On Replica test views, we achieve **68.3% mIoU** with **2.06 FPS**, compared to SemGauss-SLAM's 88.62% mIoU at significantly lower frame rates. This trade-off reflects our design priority: end-to-end interactive throughput for online robotic mapping rather than offline reconstruction quality. In our setting, adding semantic-guided loop closure and global optimization improves consistency under perceptual aliasing. Cross-view semantic consistency reaches 0.87, and our open-vocabulary capability (78.3% recall on novel objects) enables broader applicability beyond closed-set scenarios.

TABLE IV
RENDERING PERFORMANCE ON 3 DATASETS. BOLD NUMBERS DENOTE BEST RESULTS.

Dataset	Replica			TUM			ScanNet		
Method	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
NICE-SLAM [13]	24.42	0.892	0.233	14.86	0.614	0.441	17.54	0.621	0.548
ESLAM [19]	28.06	0.923	0.245	15.26	0.478	0.569	15.29	0.658	0.488
Point-SLAM [20]	35.17	0.975	0.124	16.62	0.696	0.256	19.82	0.751	0.514
Loopy-SLAM [6]	35.47	**0.981**	0.109	12.94	0.489	0.645	15.23	0.663	0.671
SplaTAM [8]	34.11	0.970	0.100	22.80	**0.893**	**0.178**	19.14	0.716	0.358
LDO-SLAM (Ours)	**37.72**	0.976	**0.08**	**24.28**	0.860	0.22	**25.76**	**0.837**	**0.325**

TABLE V
RUNTIME ANALYSIS ON REPLICA ROOM0. BOLD: BEST RESULT FOR EACH COLUMN.

Method	ATE (cm) ↓	Tracking (s/frame) ↓	Mapping (s/frame) ↓	FPS ↑
Point-SLAM [20]	0.61	8.20	40.8	0.06
SplaTAM [8]	0.31	3.2	5.3	0.12
MonoGS [9]	0.47	1.47	3.05	0.45
LDO-SLAM (Ours)	**0.25**	**0.08**	**0.35**	**2.06**

TABLE VI
KEY HYPERPARAMETERS

Parameter	**Description**	**Value**
θ_{IoU}	Semantic association threshold	0.3
β	Bayesian fusion sharpness	2.0
θ_{loop}	Loop closure threshold	0.75
λ_{sem}	Semantic observation weight	0.5

4) Comparison with Latest Methods: LoopSplat achieves globally consistent geometry through 3DGS registration but lacks semantic understanding. RGBDS-SLAM excels in offline reconstruction quality (PSNR 42.46) with ground-truth semantic supervision but requires pre-annotated labels. Our method combines open-vocabulary semantic reasoning with end-to-end throughput (2.06 FPS on RTX 3090), trading offline reconstruction quality for online robotic mapping with semantic loop closure (85ms verification).

5) Open-Vocabulary Generalization: On a custom dataset with 15 non-COCO objects (fire extinguisher, whiteboard, projector), VLM-enhanced detection achieves 78.3% recall vs. 12.1% for YOLO-only. Cross-domain evaluation on outdoor (KITTI) and industrial scenes shows competitive localization (ATE degradation $< 15\%$ in our test) but reduced semantic accuracy (54.2% mIoU) due to domain shift, indicating room for improvement in extreme cross-domain scenarios.

6) Parameter Sensitivity Analysis: Results show stable performance under $\pm 25\%$ parameter variations on Replica/room0. $\theta_{\text{IoU}} = 0.3$ balances precision/recall, $\beta = 2.0$ provides appropriate fusion sharpness, and $\sigma_{\text{cons}} = 0.05$m aligns with indoor object boundary scales.

7) Computational Efficiency Analysis: VLM inference (120ms) runs asynchronously when YOLO confidence < 0.6. In typical scenes, VLM is invoked on $\sim$10% of frames ($\sim$12ms overhead); in complex scenes with novel categories, frequency increases to $\sim$25% (30ms overhead, FPS drops to 1.85). Factor graph optimization (200ms) triggers upon loop closure ($\sim$50 frames). Adaptive marginalization of high-entropy landmarks ($H > 0.5$) reduces optimization frequency by 40% vs. fixed intervals while maintaining similar accuracy in our evaluation.

TABLE VII
COMPARISON WITH LATEST 3DGS-SLAM METHODS (REPLICA ROOM0)

Method	ATE (cm)	PSNR	mIoU (%)	FPS
LoopSplat [10]	0.28	33.07	–	0.83
RGBDS-SLAM [11]	0.50	42.46	92.67	29.55
LDO-SLAM (Ours)	**0.25**	37.7	68.3	2.06

TABLE VIII
PARAMETER SENSITIVITY (ATE IN CM ON REPLICA ROOM0)

Parameter	0.5×	0.75×	**Default**	1.25×	1.5×
θ_{IoU}	0.28	0.26	**0.25**	0.27	0.31
β	0.27	0.26	**0.25**	0.26	0.28
σ_{cons}	0.29	0.27	**0.25**	0.26	0.27
w_{sem}	0.26	0.25	**0.25**	0.26	0.28

8) Runtime Analysis: Runtime profiling on Replica/room0 (Table 5) shows tracking in **0.084 s/frame** and mapping in **0.35 s/frame**, yielding **2.06 FPS** end-to-end (*interactive-rate* performance suitable for robotics, though below 10–30 Hz of tracking-only systems). All timings are averaged over the evaluated sequence after a short warm-up, and the reported FPS measures end-to-end throughput including tracking, mapping, and semantic assignment; asynchronous modules (VLM and global optimization) are reported with their amortized per-frame cost based on trigger frequency. Main costs: YOLO (15ms), semantic assignment (8ms), Gaussian tracking (84ms), mapping (350ms) per frame; VLM (120ms) asynchronously every 10 frames; loop detection (5ms coarse, 80ms fine) and optimization (200ms) triggered periodically.

C. Ablation Study

Adaptive fusion achieves best ATE (0.19cm) and consistency (0.89), outperforming YOLO-only (0.31cm, 0.78) and fixed fusion (0.26cm, 0.85). Our two-stage loop closure

TABLE IX
COMPONENT-WISE ABLATION ON REPLICA (ATE IN CM, PSNR IN DB)

Configuration	ATE ↓	PSNR ↑	FPS ↑
Baseline (no semantic)	0.40	35.2	2.8
+ Semantic fusion	0.31	36.1	2.3
+ Semantic loop closure	0.28	36.8	2.1
Full (Ours)	**0.25**	**37.7**	2.06

achieves 94.7% precision and 88.3% recall vs. geometric-only (78.3%) and semantic-only (82.1%).

V. CONCLUSION

We presented LDO-SLAM, a dense SLAM framework integrating semantic enhancement and robust loop closure within 3DGS, designed for *interactive-rate robotic applications*. By unifying semantic embedding and hierarchical optimization, LDO-SLAM achieves a compelling balance between semantic understanding and computational efficiency. Comprehensive experiments demonstrate: (1) interactive-rate performance (2.06 FPS) with competitive semantic accuracy, prioritizing real-time applicability over offline reconstruction quality; (2) faster loop closure speed through semantic guidance; (3) unique open-vocabulary capability beyond closed-set methods; (4) efficient VLM integration.

Unlike offline semantic reconstruction systems, LDO-SLAM targets real-time robotic navigation and manipulation, where interactive-rate feedback and global consistency are critical. *Limitations:* Our open-vocabulary semantics may be affected by VLM misclassification and label-space mapping ambiguity, and performance can degrade under severe domain shift or fast motion. Future work will explore lightweight VLM distillation, stronger temporal calibration, and enhanced cross-domain transfer to further improve efficiency and generalization in diverse real-world environments.

REFERENCES

[1] Carlos Campos, Richard Elvira, Juan J Gómez Rodríguez, José MM Montiel, and Juan D Tardós, "Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam," *IEEE transactions on robotics*, vol. 37, no. 6, pp. 1874–1890, 2021.

[2] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," *Communications of the ACM*, vol. 65, no. 1, pp. 99–106, 2021.

[3] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis, "3d gaussian splatting for real-time radiance field rendering.," *ACM Trans. Graph.*, vol. 42, no. 4, pp. 139–1, 2023.

[4] Huajian Huang, Longwei Li, Hui Cheng, and Sai-Kit Yeung, "Photo-slam: Real-time simultaneous localization and photorealistic mapping for monocular stereo and rgb-d cameras," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 21584–21593.

[5] Youmin Zhang, Fabio Tosi, Stefano Mattoccia, and Matteo Poggi, "Go-slam: Global optimization for consistent 3d instant reconstruction," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 3727–3737.

[6] Lorenzo Liso, Erik Sandström, Vladimir Yugay, Luc Van Gool, and Martin R Oswald, "Loopy-slam: Dense neural slam with loop closures," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 20363–20373.

[7] Zachary Teed and Jia Deng, "Droid-slam: Deep visual slam for monocular, stereo, and rgb-d cameras," *Advances in neural information processing systems*, vol. 34, pp. 16558–16569, 2021.

[8] Nikhil Keetha, Jay Karhade, Krishna Murthy Jatavallabhula, Gengshan Yang, Sebastian Scherer, Deva Ramanan, and Jonathon Luiten, "Splatam: Splat track & map 3d gaussians for dense rgb-d slam," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 21357–21366.

[9] Hidenobu Matsuki, Riku Murai, Paul HJ Kelly, and Andrew J Davison, "Gaussian splatting slam," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 18039–18048.

[10] Liyuan Zhu, Yue Li, Erik Sandström, Shengyu Huang, Konrad Schindler, and Iro Armeni, "Loopsplat: Loop closure by registering 3d gaussian splats," *arXiv preprint arXiv:2408.10154*, 2024.

[11] Zhenzhong Cao, Chenyang Zhao, Qianyi Zhang, Jinzheng Guang, Yinuo Song, and Jingtai Liu, "Rgbds-slam: A rgb-d semantic dense slam based on 3d multi level pyramid gaussian splatting," *IEEE Robotics and Automation Letters*, 2025.

[12] Edgar Sucar, Shikun Liu, Joseph Ortiz, and Andrew J Davison, "imap: Implicit mapping and positioning in real-time," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 6229–6238.

[13] 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*, 2022, pp. 12786–12796.

[14] Siting Zhu, Renjie Qin, Guangming Wang, Jiuming Liu, and Hesheng Wang, "Semgauss-slam: Dense semantic gaussian splatting slam," *arXiv preprint arXiv:2403.07494*, 2024.

[15] Mingrui Li, Shuhong Liu, Heng Zhou, Guohao Zhu, Na Cheng, Tianchen Deng, and Hongyu Wang, "Sgs-slam: Semantic gaussian splatting for neural dense slam," in *European Conference on Computer Vision*. Springer, 2024, pp. 163–179.

[16] 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," *arXiv preprint arXiv:2503.01646*, 2025.

[17] Tomas Berriel Martins, Martin R Oswald, and Javier Civera, "Ovo-slam: Open-vocabulary online simultaneous localization and mapping," *arXiv preprint arXiv:2411.15043*, 2024.

[18] Siting Zhu, Guangming Wang, Hermann Blum, Jiuming Liu, Liang Song, Marc Pollefeys, and Hesheng Wang, "Sni-slam: Semantic neural implicit slam," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 21167–21177.

[19] 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*, 2023, pp. 17408–17419.

[20] 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*, 2023, pp. 18433–18444.

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

[22] Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, et al., "The replica dataset: A digital replica of indoor spaces," *arXiv preprint arXiv:1906.05797*, 2019.

[23] Jürgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers, "A benchmark for the evaluation of rgb-d slam systems," in *2012 IEEE/RSJ international conference on intelligent robots and systems*. IEEE, 2012, pp. 573–580.

[24] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner, "Scannet: Richly-annotated 3d reconstructions of indoor scenes," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 5828–5839.

[25] Glenn Jocher et al., "Yolov5," https://github.com/ultralytics/yolov5, 2020.