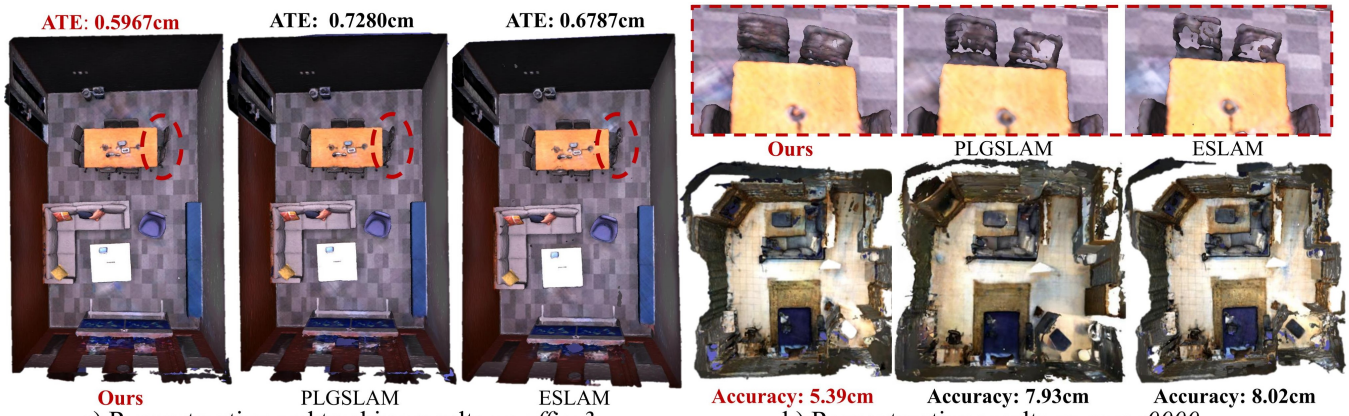
HS-SLAM: Hybrid Representation with Structural Supervision for Improved Dense SLAM

Ziren Gong[1] Fabio Tosi[1] Youmin Zhang[2] Stefano Mattoccia[1] Matteo Poggi[1]
[1]University of Bologna [2]Rock Universe

Project page: https://zorangong.github.io/HS-SLAM/



a) Reconstruction and tracking results on *office3* b) Reconstruction results on *scene0000*

Fig. 1. **Trajectory errors on *Office3* (Replica [1]) and 3D Reconstruction results on *scene0000* (ScanNet [2])**. Our framework yields the accurate 3D reconstruction shown in a), together with more precise camera tracking. Compared to existing methods such as PLGSLAM and ESLAM, HS-SLAM better preserves details (see red circles and boxes), achieving superior reconstruction accuracy, as shown in b).

***Abstract*— NeRF-based SLAM has recently achieved promising results in tracking and reconstruction. However, existing methods face challenges in providing sufficient scene representation, capturing structural information, and maintaining global consistency in scenes emerging significant movement or being forgotten. To this end, we present HS-SLAM to tackle these problems. To enhance scene representation capacity, we propose a hybrid encoding network that combines the complementary strengths of hash-grid, tri-planes, and one-blob, improving the completeness and smoothness of reconstruction. Additionally, we introduce structural supervision by sampling patches of non-local pixels rather than individual rays to better capture the scene structure. To ensure global consistency, we implement an active global bundle adjustment (BA) to eliminate camera drifts and mitigate accumulative errors. Experimental results demonstrate that HS-SLAM outperforms the baselines in tracking and reconstruction accuracy while maintaining the efficiency required for robotics.**

I. Introduction

Visual simultaneous localization and mapping (SLAM) has been widely applied in robotics. In the last decades, numerous traditional SLAM frameworks have achieved camera tracking and sparse maps, with hand-crafted methods [3]–[10] demonstrating impressive scalability and fast speed. Deep-learning [11]–[13] frameworks further enhance the accuracy and robustness of pose estimation. However, robots need dense reconstruction capacity with various applications of indoor reconstruction and virtual/augmented reality [14]. With the emergence of NeRF [15] and 3D Gaussian Splatting (3DGS) [16], recent attention has turned to both NeRF-centric and 3DGS-centric SLAM.

3DGS-centric SLAM represents scenes with explicit 3D Gaussian-shaped primitives. SplaTAM [17] is the first open-source work to use the 3DGS [16] in SLAM. GS-SLAM [18] also works out reasonably dense maps with 3D GS. Although these works demonstrate promising reconstruction capability, they face limitations concerning their huge memory usage and slow speed when handling both tracking and mapping simultaneously [19]. These drawbacks hinder their applicability in robotics, where real-time performance and efficiency are essential.

Compared to 3DGS, NeRF-centric SLAM offers a more compact scene representation and requires significantly less GPU memory [20], making it highly suitable for resource-constrained robots. iMAP [21] is the first work to represent scenes with implicit neural features. NICE-SLAM [22], ESLAM [23], and Co-SLAM [24] further explore alternative representations to reconstruct high-fidelity scenes. Recently, the SOTA method PLGSLAM [25] further enhances the reconstruction accuracy in large-scale indoor scenes.

In this context, we investigate whether NeRF-centric SLAM has achieved its optimal potential from the following perspectives: i) scene representation, ii) structural supervision, and iii) global consistency. In terms of the former, the structure of embeddings plays a crucial role [26]. Different encodings have distinct advantages: Tri-planes [27] enable rapid convergence and enhance the completeness of unobserved regions, while hash grids [28] excel at capturing details. Regarding supervision, radiance-field-based approaches only learn from individual pixels, neglecting potential struc-

tural information in scenes, while for what concerns global consistency, the current NeRF-centric methods struggle with scenes being forgotten or in the occurrence of significant movements.

To this end, we propose HS-SLAM, a framework marrying a Hybrid representation with Structural supervision for dense RGB-D SLAM. The former leverages the complementary advantages of hash grid [28], tri-planes [27] and one-blob [29], achieving superior scene reconstruction. For the latter we incorporate patch rendering loss, enabling HS-SLAM to better capture structural features and perceive changes in unobserved regions. Finally, we introduce an active global BA to allocate more samples to historical observation where new regions emerge or scenes are being forgotten. This avoids re-training well-optimized scenes while eliminating camera drift and mitigating accumulative errors.

To resume, HS-SLAM envelops the following properties:

- **Enhanced scene representation capacity:** We introduce a novel hybrid representation to exploit complementary strengths of encodings, representing entire scenes with both completeness and detailed textures.
- **Non-local spatial and structural supervision:** We incorporate patch rendering loss for structural supervision, sampling patches of non-local pixels to capture structural information in scenes.
- **Global consistency:** Our framework proposes an active global BA strategy. It optimizes our SLAM system using historical observations while actively sampling from frames with new regions or scenes being forgotten.

II. Related Work

We briefly review the literature relevant to our work, referring to [30] for a detailed overview of the latest advances.

A. Traditional SLAM Frameworks

These approaches generally refer to traditional visual SLAM pipelines, or to the deep-learning-based SLAM [11]–[13] – usually in a frame-to-frame fashion based on the visual features extracted from history frames. Furthermore, this kind of method commonly composes tracking, mapping, global BA, and loop closure. Traditional SLAM frameworks using depth points [4], [6], [31], surfels [32], and volumetric representations [33] achieve globally consistent reconstruction. However, their sparse and limited representation suffers from undesirable dense reconstruction results.

B. NeRF-centric Frameworks

These approaches, also known as *neural* SLAM systems, estimate camera poses and reconstruct dense maps by modeling scenes within MLPs and optimizing the scene representation. iMAP [21] and Nice-SLAM [22] are pioneers in bringing implicit neural mapping to SLAM. Point-SLAM [34] and Loopy-SLAM [35] introduce a novel neural point embedding for dense reconstruction, allowing for the flexibility to correct and adjust local maps, yet with slow processing prohibiting deployment in robotics. ESLAM [23] and Co-SLAM [24] explore, respectively, tri-planes and hash grid embeddings for scene representation, improving both processing speed and reconstruction accuracy, with PLGSLAM [25] further improving the representation capacity in large indoor scenes. GO-SLAM [36] and KN-SLAM [37] exploit external trackers to achieve a good trade-off between camera poses and scene reconstruction. Nevertheless, existing NeRF-centric methods still underutilize both the power of features representations [26] and the structural supervision available from color images [38], thus attaining sub-optimal performance both at tracking and mapping.

C. 3DGS-centric Framworks

These systems exploit explicit 3D Gaussian Splatting [16] to represent the whole scene, iteratively growing the mapped area and offering the flexibility to adjust the reconstructed regions locally. SplaTAM [17] is the first open-source pioneer on this track, achieving high-quality color and depth rendering followed by GS-SLAM [18]. Although the 3DGS-centric frameworks prove the potential of 3DGS for SLAM applications, they are memory-hungry and run at a much lower speed, preventing their deployment in robotics.

III. Method

We introduce HS-SLAM, whose architecture is depicted in Fig. 2, from the following aspects: hybrid scene representation (III-A), SDF-based rendering and structural supervision (III-B), and active global bundle adjustment (III-C).

A. Hybrid Scene Representation

While [23], [24], and [25] further explore alternative representations, they rely solely on a single type of parameter encoding. However, each parameter encoding offers unique strengths: hash grids excel at capturing fine details while tri-planes effectively represent unobserved segments and enable rapid convergence [26]. This insight naturally leads to the idea of combining their complementary advantages to achieve stronger scene representation. We propose a hybrid scene representation to improve the accuracy and completeness of reconstruction further. Hash grids and tri-planes are utilized as parametric encodings to capture high-frequency features [23], [24] like color and geometry details, while the one-blob technique is employed as a coordinate embedding to encode low-frequency information [24], such as coherence and smoothness priors.

Specifically, we employ a multi-resolution hash grid to encode various spatial features, capturing more detailed textures. Tri-linear interpolation is then applied to query the hash grid feature $\nu_\alpha(x_i)$. For low-frequency features, we adopt the one-blob to encode spatial coordinates. The one-blob feature $\gamma(x_i)$ encodes the smoothness and coherence priors to achieve high fidelity. To enhance stability in geometry reconstruction [23], two distinct tri-planes are designed at both coarse-level $\tau^c(x_i)$ and fine-level $\tau^f(x_i)$, capturing appearance $f^a_\phi(*)$ and geometry features $f^g_\omega(*)$ separately.

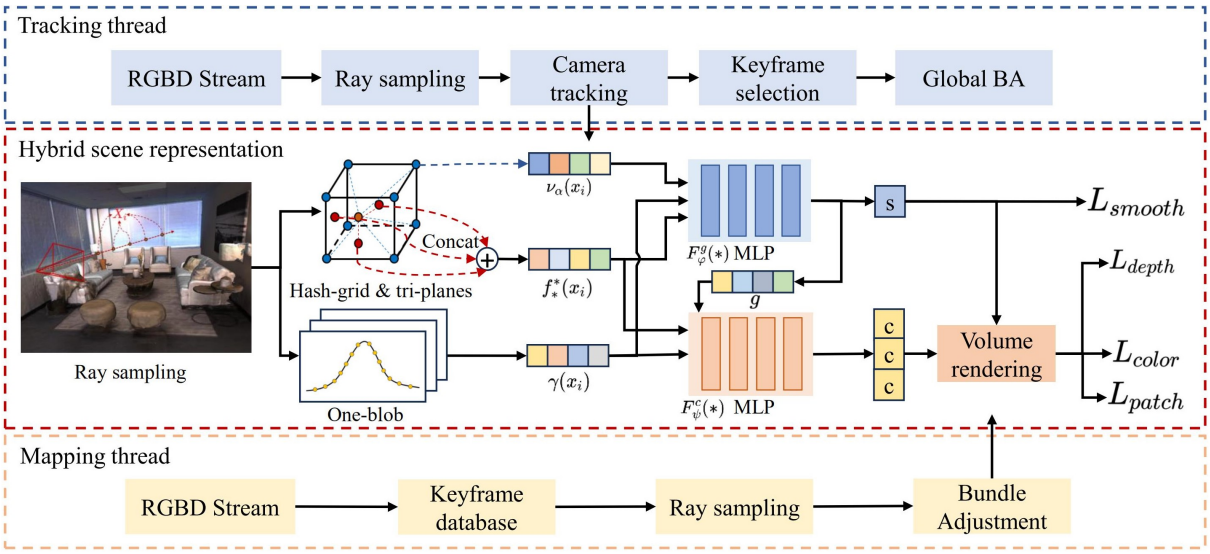


Fig. 2: **The overview of HS-SLAM.** 1) Tracking thread: The proposed global BA actively optimizes our system taking ray sampling from historical observations. 2) Mapping thread: The system optimizes hybrid encodings by performing bundle adjustment. 3) Hybrid scene representation: We carefully designed the encoders composed of hash grid, tri-planes, and one-blob to improve completeness and accuracy. Meanwhile, the proposed structural supervision (patch rendering loss) allows the network to capture global scene structures.

The process of tri-plane encoding is formulated as follows:

$$\begin{aligned} f_a^c(x_i) &= \tau_{a-xy}^c(x_i) + \tau_{a-xz}^c(x_i) + \tau_{a-yz}^c(x_i) \\ f_a^f(x_i) &= \tau_{a-xy}^f(x_i) + \tau_{a-xz}^f(x_i) + \tau_{a-yz}^f(x_i) \\ f_\phi^a(x_i) &= Concat\left(f_a^c, f_a^f\right) \\ f_g^c(x_i) &= \tau_{g-xy}^c(x_i) + \tau_{g-xz}^c(x_i) + \tau_{g-yz}^c(x_i) \\ f_g^f(x_i) &= \tau_{g-xy}^f(x_i) + \tau_{g-xz}^f(x_i) + \tau_{g-yz}^f(x_i) \\ f_\omega^g(x_i) &= Concat\left(f_g^c, f_g^f\right) \end{aligned} \tag{1}$$

where, f_a^c, f_a^f, f_g^c, f_g^f represent the coarse and fine level of appearance features f_ϕ^a and geometry features f_ω^g captured by tri-planes. The coordinates of the sampled points are denoted as x_i. $\left[\tau_{a-xy}^*(x_i), \tau_{a-xz}^*(x_i), \tau_{a-yz}^*(x_i)\right]$ and $\left[\tau_{g-xy}^*(x_i), \tau_{g-xz}^*(x_i), \tau_{g-yz}^*(x_i)\right]$ correspond to the three feature planes at both coarse and fine levels for appearance and geometry features, respectively. The two are then concatenated to form the final appearance and geometry features.

To learn the geometry features, our network combines the hash grid features $\nu_\alpha(x_i)$, tri-planes features $f_\omega^g(x_i)$, and one-blob features $\gamma(x_i)$. A shallow two-layer MLP F_φ^g serves as the geometry decoder, which processes the hybrid encoding to predict signed distance function (SDF) $\phi_g(x_i)$ and a geometry embedding g:

$$F_\varphi^g\left(\nu_\alpha(x_i), f_\omega^g(x_i), \gamma(x_i)\right) \rightarrow \left(g, \phi_g(x_i)\right) \tag{2}$$

Then, a color decoder F_ψ^c inputs the appearance feature f_ϕ^a, latent embedding g, and one-blob features $\gamma(x_i)$ to predict the raw color $\phi_a(x_i)$, with f_ϕ^a easing convergence, g capturing detailed texture information, and $\gamma(x_i)$ enhancing the smoothness and coherence of the predicted results.

$$F_\psi^c\left(f_\phi^a(x_i), \gamma(x_i), g\right) \rightarrow \phi_a(x_i) \tag{3}$$

B. SDF-based Rendering and Structural Supervision

SDF-based Rendering. Our framework emulates the ray-casting process used in NeRF [15]. We implement different pixel sampling strategies for selecting rays depending on the specific component. In the global BA, detailed in (III-C), the network randomly samples pixels and their corresponding rays from historical observations. In the mapping thread, pixels are sampled from a window of frames, which includes the current frame, the adjacent frame, and additional frames randomly selected from keyframes. For all sampled points along the rays $\{p_n\}_{n=1}^N$, we query TSDF $\phi_g(p_n)$ and raw color $\phi_a(p_n)$ from our implicit neural network. SDF-based rendering [39] is then applied to calculate volume densities:

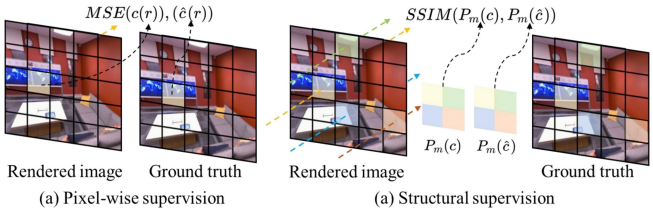
$$\sigma(p_n) = \beta \cdot Sigmoid\left(-\beta \cdot \phi_g(p_n)\right) \tag{4}$$

where $\sigma(p_n)$ denotes the volume density, and β is a learnable parameter controlling the sharpness of the surface boundary. Then, the termination probability w_n, color c, and depth d can be rendered by the calculated volume densities:

$$\begin{aligned} w_n &= exp\left(-\sum_{k=1}^{n-1}\sigma(p_k)\right)\left(1 - exp\left(-\sigma(p_n)\right)\right) \\ \text{with} \quad c &= \sum_{n=1}^{N} w_n \phi_a(p_n), \qquad d = \sum_{n=1}^{N} w_n z_n \end{aligned} \tag{5}$$

where z_n represents the depth of sampled points p_n.

Patch-Based Structural Supervision. Existing methods [23]–[25] only sample individual pixels and their corresponding rays separately. As illustrated in Fig. 3, pixel-wise supervision employs MSE losses among individual pixels. However, this approach neglects any structural information.



(a) Pixel-wise supervision (a) Structural supervision

Fig. 3: **Differences between the pixel-wise and structural supervision.** This figure illustrates the key process of the pixel-wise and structural supervision. Our structural supervision captures non-local information from scenes.

In contrast, we revise this paradigm to leverage the Structural Similarity Index Measure (SSIM) to capture structural information [38]. Specifically, our network randomly selects r pixels from the rendered set R and their corresponding ground truth R in the current frame or the keyframe database. The selected r rendered pixels and their ground truth then form patches $P\left(c\left(r\right)\right)$ and $P\left(\hat{c}\left(r\right)\right)$, with $c\left(r\right)$, $\hat{c}\left(r\right)$ denoting rendered and ground truth color.

Afterward, we use a 3×3 kernel and a stride factor of 4 to compute SSIM. The patch sampling and SSIM estimation steps are repeated M times. The patch rendering loss is computed by the average of M estimated SSIM:

$$L_{patch} = 1 - \frac{1}{M} \sum_{m=1}^{M} SSIM\left(P_m\left(\hat{c}\right), P_m\left(c\right)\right) \tag{6}$$

We also compute color and depth rendering loss L_{color}, L_{depth} between rendered values c_n, d_n and ground truth values $\hat{c}_n, \hat{d}_n$ with L_2 errors.

$$L_{color} = \frac{1}{N} \sum_{n=1}^{N} \left(\hat{c}_n - c_n\right)^2, L_{depth} = \frac{1}{N} \sum_{n=1}^{N} \left(\hat{d}_n - d_n\right)^2 \tag{7}$$

Other Losses. In addition, we also enforce a feature smoothness loss to decrease noisy representation generated by hash grid $\nu_\alpha\left(x_i\right)$ features in unobserved regions [24]:

$$L_{smooth} = \frac{1}{|X|} \sum_{x \in X} \triangle_x^2 + \triangle_y^2 + \triangle_z^2 \tag{8}$$

with X denoting a small random region to perform this loss, $\triangle_{x,y,z} = \nu_\alpha\left(x + \varepsilon_{x,y,z}\right) - \nu_\alpha\left(x\right)$ representing feature-metric differences between adjacent sampled vertices on the hash grid along the three dimensions.

We approximate the signed distance field as the following objective function:

$$L_{sdf}\left(P_r^T\right) = \frac{1}{|R|} \sum_{r \in R} \frac{1}{|P_r^T|} \sum_{p \in P_r^T} \left(\phi_g\left(p\right) \cdot T + z\left(p\right) - D\left(r\right)\right) \tag{9}$$

with P_r^T denoting a group of points along the ray that lies within the truncation region, i.e. $|z\left(p\right) - D\left(r\right)| < T$, $z\left(p\right)$ being the planar depth of point p, and $D\left(r\right)$ representing the measured ray depth. Furthermore, we distinguish the importance of points closer to the surface, situated in the middle of the truncation region P_r^{Tm}, from those located at the tail of the truncation region P_r^{Tt}:

$$L_{sdfm} = L_{sdf}\left(P_r^{Tm}\right), L_{sdft} = L_{sdf}\left(P_r^{Tt}\right) \tag{10}$$

Finally, for points sampled far from the surface $|z\left(p\right) - D\left(r\right)| \geq T$, we enforce the predicted SDF values to be the truncated distance:

$$L_{fs} = \frac{1}{|R|} \sum_{r \in R} \frac{1}{\left|P_r^{fs}\right|} \sum_{p \in P_r^{fs}} \left(\phi_g\left(p\right) - 1\right)^2 \tag{11}$$

C. Active Global Bundle Adjustment

While existing NeRF-centric frameworks employ global BA to mitigate camera drifts, they select keyframes based on a fixed stride size, overlooking frames with significant motion and large trajectory errors. Meanwhile, processing well-optimized scenes again wastes lots of memory. To address this problem, we propose an active global BA to enhance the global consistency of both tracking and reconstruction. Our global BA actively selects the frames with suboptimal trajectory poses and scene reconstruction from history observations. We observe the fact that the global loss of frames increases markedly when the camera undergoes significant motion or the network starts to forget scenes. Consequently, a keyframe is selected when the global loss exceeds a threshold $L > t_l$ (we set $t_l = 0.09$). Additionally, we aim to allocate more samples from the frames with higher losses. To this end, we rank the keyframes by loss and sample rays from the top N keyframes (with $N = 15$). This approach prioritizes the keyframes inaccurately rendered by the model, which refers to newly explored frames or frames that the network is starting to forget. By randomly sampling rays from this prioritized global keyframe database, we optimize the global history trajectory, correcting inaccurate camera poses.

IV. Experiments

We now introduce our experimental evaluation, by detailing our setting and implementation details, then discussing the results achieved over several datasets.

A. Experimental setup

We introduce the datasets used in our evaluation, as well as the frameworks we compare with, the evaluation metrics, and the implementation details of HS-SLAM.

Datasets. To evaluate the tracking and reconstruction accuracy, we test our HS-SLAM on several specific scenes in the following datasets: Replica [1], ScanNet [2], and TUM RGB-D [40]. The former is a semi-synthetic dataset allowing for the evaluation of both aspects involved in SLAM performance, whereas the others are usually employed to evaluate trajectory accuracy only.

Baselines. Considering the efficiency required in robotics applications, we focus our comparison with NeRF-centric SLAM systems achieving a reasonable framerate, such as Co-SLAM [24], ESLAM [23], and PLGSLAM [25] – the current state-of-the-art NeRF-centric framework on Replica – as the primary baselines to compare the accuracy of

TABLE I: **Reconstruction and tracking results on Replica (mesh culling strategy in [24]).** We compare our HS-SLAM with NeRF-centric SLAM systems that better meet efficiency requirements.

	Acc.(cm)	Comp.(cm)	Comp.Rate(%)	Depth l1(cm)	Mean(cm)	ATE RMSE(cm)
Co-SLAM	2.10	2.08	93.44	1.51	0.935	1.059
ESLAM	2.18	1.75	96.46	0.94	0.565	0.707
PLGSLAM	2.15	1.74	96.25	0.83	0.525	0.635
HS-SLAM (ours)	**1.96**	**1.67**	**96.57**	**0.71**	**0.463**	**0.528**
SplaTAM	-	-	-	-	-	0.36-

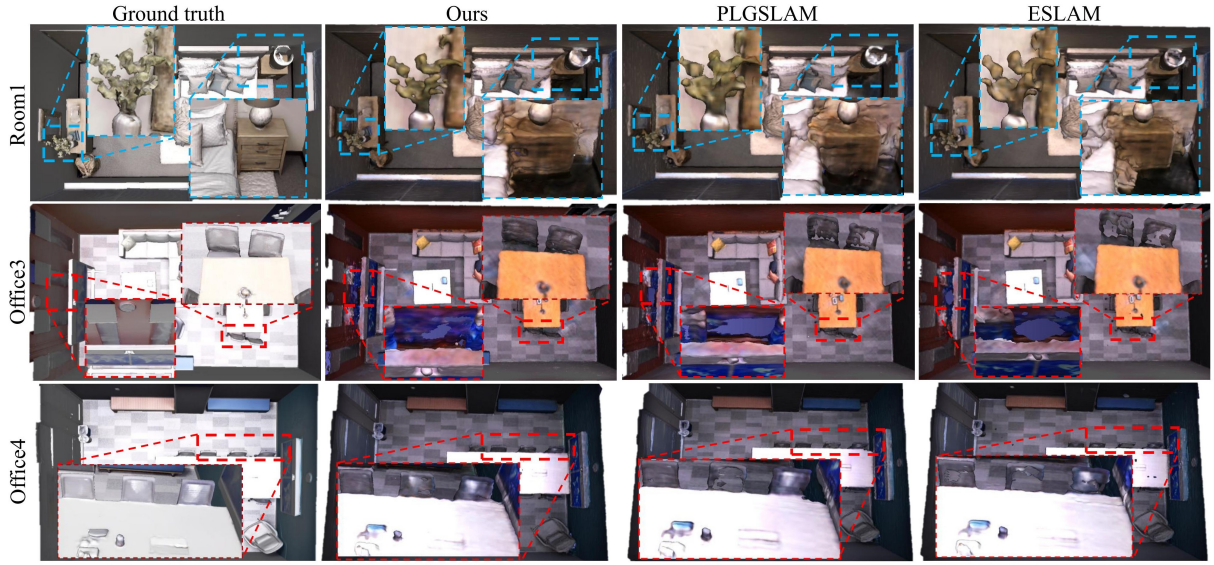


Fig. 4: **Qualitative reconstruction results on Replica.** For each scene, we provide detailed drawings, highlighting complex regions to emphasize the textures and completeness of our reconstructed scenes. HS-SLAM demonstrates superior scene representation capabilities, effectively capturing intricate textures (e.g., flowers and nightstands) and achieving higher completeness in challenging areas (e.g., chairs and walls).

reconstruction and tracking with, while reporting SplaTAM as a reference for 3DGS-centric systems. When focusing on performance analysis, we compare our HS-SLAM also with Point-SLAM [34] to demonstrate our efficiency.

Metrics. To evaluate reconstruction quality, we use standard metrics such as *depth l1* (cm), *accuracy* (cm), *completion* (cm), and *completion rate* $(\%)$. Unlike some prior works [22], [23], [25], we adopt the mesh culling strategy proposed in [24], which minimizes artifacts outside the region of interest (ROI) and preserves occluded parts within it. This strategy includes frustum culling, occlusion handling, and the use of virtual cameras, covering occluded parts within the ROI. For camera tracking, we compute *ATE RMSE* (cm).

Implementation Details. We implement HS-SLAM using PyTorch, on a desktop PC with NVIDIA RTX 3090 Ti GPU. We sample 2000 pixels for tracking and 4000 pixels for mapping. We use 16 bins for one-blob and a 16-level hash grid with a maximum voxel size of 2 cm. The tri-planes are set at 24cm/6cm and 24cm/3cm resolutions, each with 32 channels. The decoders consisted of two-layer MLPs with 32 channels in the hidden layers.

B. Results on Replica

We evaluate our HS-SLAM on Replica, using the same RGB-D sequences as the baselines to test reconstruction accuracy. Tab. I shows how HS-SLAM achieves both superior reconstruction accuracy, outperforming any NeRF-centric SLAM system using single parametric encoding, either hash grids (Co-SLAM) or tri-planes (ESLAM, PLGSLAM) thanks to the hybrid encoding. Furthermore, our global BA strategy based on active sampling allows for consistently superior tracking accuracy. Fig. 4 shows a qualitative comparison between HS-SLAM, PLGSLAM, and ESLAM, highlighting the lower presence of artifacts in our reconstructions.

C. Ablation Studies

We assess the impact of any components in HS-SLAM on the final accuracy, collecting the results in Tab. II.

Hybrid Scene Representation. We compare several embedding variants aimed at finding the one best suited for SLAM. To stress the importance of combining different representations to complement their strengths and weaknesses, at the top of the table we report results achieved by a single embedding, hash grid (A), or tri-planes (B), by doubling the number of features they store. Both fail at obtaining satisfying results across the overall set of metrics, confirming that a hybrid encoding is crucial to overcome the limitations of the single ones – and does not simply benefit from having more features. We then evaluate the results achieved using the hybrid encoding proposed in [26], claiming that dense

TABLE II: **Ablation study on Replica.** We study the impact of any of our components on mapping and tracking accuracy.

		Acc.(cm)	Comp.(cm)	Comp.Rate(%)	Depth l1(cm)	Mean(cm)	ATE RMSE(cm)
	HS-SLAM (ours)	**1.96**	**1.67**	96.57	**0.71**	**0.463**	**0.528**
(A)	Only hash grid(×2)	1.98	2.09	93.57	1.44	0.944	1.073
(B)	Only tri-planes(×2)	2.05	1.68	96.68	0.81	0.476	0.547
(C)	dense + tri-planes [26]	5.61	1.73	**97.03**	0.83	0.474	0.555
(D)	w/o hash grid	2.06	1.68	96.55	0.75	0.487	0.558
(E)	w/o tri-planes	2.11	1.93	94.52	1.21	0.851	0.915
(F)	w/o one-blob	1.98	1.68	96.53	0.74	0.472	0.539
(G)	w/o patch loss	1.97	1.67	96.51	0.73	0.476	0.547
(H)	w/o global BA	2.07	1.67	96.56	0.73	0.495	0.568
(I)	w/ Co-SLAM global BA	2.01	1.67	96.57	0.73	0.485	0.557

TABLE III: **Tracking and mapping results on ScanNet.**

	Acc.(cm)	Comp.(cm)	Comp.Rate(%)	ATE RMSE(cm)
Co-SLAM	31.31	3.83	**77.36**	9.37
ESLAM	25.96	4.27	73.58	7.52
PLGSLAM	21.14	4.29	74.78	**6.77**
HS-SLAM (ours)	**19.34**	**4.25**	73.33	7.42
SplaTAM	-	-	-	11.88

TABLE IV: **Tracking results on TUM RGB-D.**

	fr1/desk(cm)	fr2/xyz(cm)	fr3/office(cm)	Average(cm)
Co-SLAM	2.754	1.747	3.293	2.598
ESLAM	2.519	1.212	2.786	2.172
HS-SLAM (ours)	**2.485**	**1.11**	**2.77**	**2.122**
SplaTAM	3.35-	1.24-	5.16-	3.25-

grids and tri-planes allow for the best results (C). Although yielding the highest completion rate, this solution performs worse on most of the remaining metrics, confirming our synergy between hash grid, tri-planes, and one-blob encoding as the optimal choice for hybrid encoding. This is further confirmed by removing a single among these representations (D-F), which always reduces the results on any metric.

Structural Supervision. The absence of structural supervision (G) introduces a moderate drop in mapping accuracy, as well as in overall tracking quality. This proves how, despite not playing a crucial role as hybrid embeddings, the structural supervision coming from patches is necessary to obtain optimal results.

Active Global Bundle Adjustment. Removing global BA (H) yields drops both in mapping and tracking quality, highlighting its major role in reducing cumulative camera pose errors, specifically when handling newly explored scenes or scenes being forgotten. Finally, replacing it with Co-SLAM global BA (I) leads to poorer performance, demonstrating once more the advance introduced by our global BA.

D. Results on Other Datasets

ScanNet Dataset. We evaluate tracking and reconstruction accuracy on ScanNet, collecting the results in Tab. III. On the one hand, HS-SLAM outperforms other methods in reconstruction. On the other hand, PLGSLAM achieves better tracking results due to its design specifically tailored for larger scene reconstruction, a contribution orthogonal to ours. Nonetheless, HS-SLAM still demonstrates strong performance thanks to its accurate scene representation and high reconstruction completeness.

TABLE V: **Memory usage and runtime on Replica *room0***. Mem. Size sums the footprints by both encoder and decoder. Avg. FPS divides the total runtime by the number of frames.

	Scene Encoding	Mem.Size (MB)	Avg.FPS
Co-SLAM	Hash Grid + MLP	**6.72**	**7.97**
ESLAM	Feature Planes + MLP	27.21	4.62
PLGSLAM	Feature Planes + MLP	-	-
HS-SLAM (ours)	Hybrid + MLP	25.83	2.35
Point-SLAM	Neural Points + MLP	54.8/3936.8	0.23
SplaTAM	3D Gaussians	392.5	0.14

TUM Dataset. We further test HS-SLAM on the TUM dataset and evaluate tracking accuracy. Tab. IV demonstrates that our framework achieves promising tracking accuracy, outperforming once again both Co-SLAM and ESLAM.

E. Performance Analysis

We compare our HS-SLAM with the NeRF-centric SLAM systems considered so far, as well as with Point-SLAM and SplaTAM, to assess its efficiency. For a fair comparison, we evaluate the available open-source code on our desktop PC. We measured memory usage (total footprint of the encoder and decoder) and average FPS (average time required to process per frame in a sequence). Tab. V shows that Co-SLAM is the fastest method with the lowest model size, yet the least accurate according to the previous experiments. In contrast, Point-SLAM and SplaTAM demonstrate much lower speeds (less than 1 FPS) with a large model (hundreds or thousands of MB), making neural points and 3D Gaussian Splatting still unsuitable for real-time applications. HS-SLAM represents a good trade-off between the two worlds, outperforming Co-SLAM and ESLAM while retaining a few FPS in speed.

V. Conclusion

We present HS-SLAM, a dense RGB-D SLAM system that integrates hybrid scene representation with structural supervision, aided by an active global BA to mitigate camera drifts. Experimental results demonstrate HS-SLAM's superior performance in both reconstruction and pose estimation compared to existing NeRF-centric SLAM systems, achieving the best trade-off between accuracy and efficiency. While currently optimized for indoor environments, future integration of submaps and loop closure techniques could expand its capability to handle larger-scale scenes.

Acknowledgment. We sincerely thank the scholarship supported by China Scholarship Council (CSC).

REFERENCES

[1] J. Straub, T. Whelan, L. Ma, Y. Chen, E. Wijmans, S. Green, J. J. Engel, R. Mur-Artal, C. Ren, S. Verma *et al.*, "The replica dataset: A digital replica of indoor spaces," *arXiv preprint arXiv:1906.05797*, 2019.

[2] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. 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.

[3] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon, "Kinectfusion: Real-time dense surface mapping and tracking," in *2011 10th IEEE international symposium on mixed and augmented reality*. Ieee, 2011, pp. 127–136.

[4] R. Mur-Artal and J. D. Tardós, "Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras," *IEEE transactions on robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.

[5] X. Li, S. Lin, M. Xu, D. Dai, and J. Wang, "Po-slam: A novel monocular visual slam with points and objects," in *2021 4th International conference on artificial intelligence and big data (ICAIBD)*. IEEE, 2021, pp. 454–458.

[6] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. Montiel, and J. 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.

[7] X. Li, D. Liu, and J. Wu, "Cto-slam: Contour tracking for object-level robust 4d slam," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 9, 2024, pp. 10 323–10 331.

[8] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison, "Dtam: Dense tracking and mapping in real-time," in *2011 international conference on computer vision*. IEEE, 2011, pp. 2320–2327.

[9] R. F. Salas-Moreno, R. A. Newcombe, H. Strasdat, P. H. Kelly, and A. J. Davison, "Slam++: Simultaneous localisation and mapping at the level of objects," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2013, pp. 1352–1359.

[10] T. Whelan, S. Leutenegger, R. F. Salas-Moreno, B. Glocker, and A. J. Davison, "Elasticfusion: Dense slam without a pose graph." in *Robotics: science and systems*, vol. 11. Rome, Italy, 2015, p. 3.

[11] K. Tateno, F. Tombari, I. Laina, and N. Navab, "Cnn-slam: Real-time dense monocular slam with learned depth prediction," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 6243–6252.

[12] Y. Li, N. Brasch, Y. Wang, N. Navab, and F. Tombari, "Structure-slam: Low-drift monocular slam in indoor environments," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6583–6590, 2020.

[13] Z. Teed and J. Deng, "Droid-slam: Deep visual slam for monocular, stereo, and rgb-d cameras," *Advances in neural information processing systems*, vol. 34, pp. 16 558–16 569, 2021.

[14] B. Tang and S. Cao, "A review of vslam technology applied in augmented reality," in *IOP Conference Series: Materials Science and Engineering*, vol. 782, no. 4. IOP Publishing, 2020, p. 042014.

[15] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," *Communications of the ACM*, vol. 65, no. 1, pp. 99–106, 2021.

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

[17] N. Keetha, J. Karhade, K. M. Jatavallabhula, G. Yang, S. Scherer, D. Ramanan, and J. 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. 21 357–21 366.

[18] C. Yan, D. Qu, D. Xu, B. Zhao, Z. Wang, D. Wang, and X. Li, "Gs-slam: Dense visual slam with 3d gaussian splatting," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 19 595–19 604.

[19] F. Tosi, Y. Zhang, Z. Gong, E. Sandström, S. Mattoccia, M. R. Oswald, and M. Poggi, "How nerfs and 3d gaussian splatting are reshaping slam: a survey," *arXiv preprint arXiv:2402.13255*, vol. 4, 2024.

[20] S. He, Z. Osman, and P. Chaudhari, "From nerfs to gaussian splats, and back," *arXiv preprint arXiv:2405.09717*, 2024.

[21] E. Sucar, S. Liu, J. Ortiz, and A. 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.

[22] Z. Zhu, S. Peng, V. Larsson, W. Xu, H. Bao, Z. Cui, M. R. Oswald, and M. Pollefeys, "Nice-slam: Neural implicit scalable encoding for slam," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 12 786–12 796.

[23] M. M. Johari, C. Carta, and F. 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. 17 408–17 419.

[24] H. Wang, J. Wang, and L. Agapito, "Co-slam: Joint coordinate and sparse parametric encodings for neural real-time slam," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 13 293–13 302.

[25] T. Deng, G. Shen, T. Qin, J. Wang, W. Zhao, J. Wang, D. Wang, and W. Chen, "Plgslam: Progressive neural scene represenation with local to global bundle adjustment," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 19 657–19 666.

[26] T. Hua and L. Wang, "Benchmarking implicit neural representation and geometric rendering in real-time rgb-d slam," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 21 346–21 356.

[27] E. R. Chan, C. Z. Lin, M. A. Chan, K. Nagano, B. Pan, S. De Mello, O. Gallo, L. J. Guibas, J. Tremblay, S. Khamis *et al.*, "Efficient geometry-aware 3d generative adversarial networks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 16 123–16 133.

[28] T. Müller, A. Evans, C. Schied, and A. Keller, "Instant neural graphics primitives with a multiresolution hash encoding," *ACM transactions on graphics (TOG)*, vol. 41, no. 4, pp. 1–15, 2022.

[29] T. Müller, B. McWilliams, F. Rousselle, M. Gross, and J. Novák, "Neural importance sampling," *ACM Transactions on Graphics (ToG)*, vol. 38, no. 5, pp. 1–19, 2019.

[30] F. Tosi, Y. Zhang, Z. Gong, E. Sandström, S. Mattoccia, M. R. Oswald, and M. Poggi, "How nerfs and 3d gaussian splatting are reshaping slam: a survey," 2024. [Online]. Available: https://arxiv.org/abs/2402.13255

[31] J. Czarnowski, T. Laidlow, R. Clark, and A. J. Davison, "Deepfactors: Real-time probabilistic dense monocular slam," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 721–728, 2020.

[32] T. Schops, T. Sattler, and M. Pollefeys, "Bad slam: Bundle adjusted direct rgb-d slam," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 134–144.

[33] A. Dai, M. Nießner, M. Zollhöfer, S. Izadi, and C. Theobalt, "Bundlefusion: Real-time globally consistent 3d reconstruction using on-the-fly surface reintegration," *ACM Transactions on Graphics (ToG)*, vol. 36, no. 4, p. 1, 2017.

[34] E. Sandström, Y. Li, L. Van Gool, and M. R. Oswald, "Point-slam: Dense neural point cloud-based slam," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 18 433–18 444.

[35] L. Liso, E. Sandström, V. Yugay, L. Van Gool, and M. 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. 20 363–20 373.

[36] Y. Zhang, F. Tosi, S. Mattoccia, and M. 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.

[37] X. Wu, Z. Liu, Y. Tian, Z. Liu, and W. Chen, "Kn-slam: Keypoints and neural implicit encoding slam," *IEEE Transactions on Instrumentation and Measurement*, vol. 73, pp. 1–12, 2024.

[38] Z. Xie, X. Yang, Y. Yang, Q. Sun, Y. Jiang, H. Wang, Y. Cai, and M. Sun, "S3im: Stochastic structural similarity and its unreasonable effectiveness for neural fields," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 18 024–18 034.

[39] R. Or-El, X. Luo, M. Shan, E. Shechtman, J. J. Park, and I. Kemelmacher-Shlizerman, "Stylesdf: High-resolution 3d-consistent image and geometry generation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 13 503–13 513.

[40] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. 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.