ICRA2026 Late Breaking Results Poster presented at
2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

Unified Neural Gaussian SLAM with Feature Splatting

Xuyang Tang, Henry K. Chu, *Member, IEEE*, and Yuxiang Sun, *Member, IEEE*

***Abstract*—Recent advances in 3D Gaussian Splatting (3DGS) have demonstrated impressive progress in high-fidelity scene reconstruction within visual SLAM. However, existing approaches often suffer from scene inconsistency, leading to visual artifacts, and the explicit maintenance of millions of Gaussians imposes significant storage overhead. To address these limitations, we present a unified Neural Gaussian SLAM with feature splatting, which represents the spatial scene as a coherent feature space while encoding view direction, distance, and position into neural Gaussians. Arbitrary image modalities—including color, depth, normals, semantics, and even language—can be decoded from this feature space. Extensive evaluations on several challenging datasets show that our method achieves state-of-the-art performance in rendering quality, reconstruction accuracy, and pose estimation.**

***Index Terms*—3D Gaussian Splatting, RGB-D, Visual SLAM, Reconstruction**

I. Extended Abstract

Simultaneous Localization and Mapping (SLAM) plays a pivotal role in applications such as autonomous driving, robotic navigation, and 3D reconstruction. With the rapid development of Virtual Reality (VR) and Augmented Reality (AR), the demand for immersive and highly realistic experiences has been steadily increasing. Scene representations in SLAM have evolved from point clouds, surfels, voxels, and meshes to implicit radiance fields, and most recently to 3D Gaussian Splatting. High-fidelity, texture-rich reconstruction has become an inevitable trend in this development.

However, current 3DGS SLAM approaches reconstruct Gaussian scenes in which primitives are relatively independent, leading to insufficient continuity and consistency. Consequently, rendering artifacts emerge and novel view rendering suffers from poor quality. Moreover, the rendering of color, depth, normal, semantics and language embeddings is performed separately, which neglects their inherent correlations. Moreover, the explicit maintenance of millions of Gaussians results in severe memory consumption.

To address the above challenges, this poster presents Unified Neural Gaussian SLAM with Feature Splatting, whose contributions are summarized as follows:

- We present Feature Splatting SLAM, a unified 3DGS SLAM system that reconstructs a latent feature space to support decoding and rendering of diverse image data.
- We encode the parameters of Gaussians into continuous multi-scale feature planes with view direction, distance, and positional encoding information.
- We conduct extensive experiments across various datasets that demonstrate the state of the art (SOTA) performance in rendering quality, meshing, and tracking accuracy.
- Extensive evaluations across various datasets demonstrate the superior performance in rendering quality, reconstruction precision and tracking accuracy.

Specifically, as illustrated in Fig. 1 of the poster, the whole framework is divided into two threads: tracking and mapping. The RGB-D camera captures a sequence of color and depth images, which is fed into the front end for tracking. The depth image is back-projected to obtain a local point cloud, and the camera pose is estimated by registering it with the global point cloud using GICP algorithm. Scene representation is achieved via multi-scale feature planes. Specifically, coherent features are extracted from point clouds through trilinear interpolation, while positional encoding of spatial coordinates, viewing directions, and distances information produces high-dimensional feature embeddings. The concatenated features are first decoded through several multilayer perceptrons (MLPs) to obtain parameters such as the scale, quaternion, and opacity of the corresponding Gaussians at each point, thereby generating 3D Gaussians integrated with feature embeddings. Subsequently, Feature Splatting is applied for rasterization to obtain the feature image. From the high-dimensional feature image, multiple decoders are utilized to reconstruct the coherent outputs, including color, depth, and surface normal. Supervision is provided by the ground-truth data, from which the loss is computed to optimize the feature planes. In the back-end, multi-view constraints are imposed by optimizing both the point cloud and the camera poses through point-cloud reprojection error. The proposed framework is easily extensible to semantic and language rendering, since the addition of appropriate decoders enables the extraction of corresponding outputs directly from the feature image.

We evaluate our method on several public datasets: Replica and TUM-RGBD dataset. Replica is a synthetic dataset consisting of eight scenes, which offers precise color and depth measurements. The TUM-RGBD dataset, which provides accurate camera poses from motion capture systems, is a standard benchmark for evaluating SLAM tracking accuracy, despite its low image quality and noisy depth measurements. Our method is evaluated against advanced SLAM approaches reproduced using official implementations, comprising three NeRF-based methods (NICE-SLAM, Co-SLAM, and ESLAM) and two 3DGS-based methods (SplaTAM and MonoGS). We present the Root Mean Square Error (RMSE) of

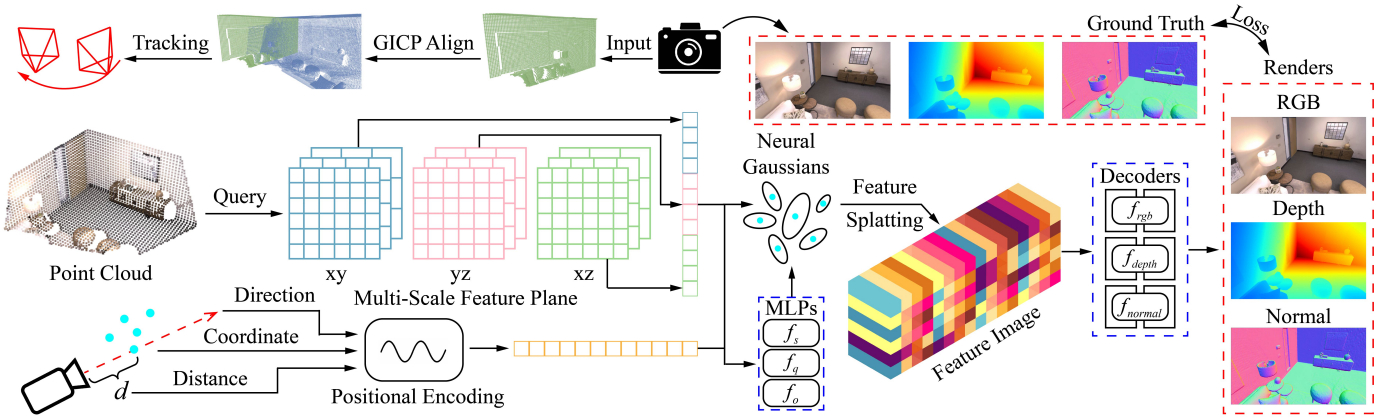

Fig. 1. The pipeline of our feature splatting SLAM.

the average Absolute Trajectory Error (ATE) to evaluate tracking accuracy. For rendering quality, we employ three standard photometric metrics: PSNR, SSIM, and LPIPS. Reconstructed meshes are evaluated using Precision, Recall, and F1-score (<1 cm, %), as well as the Depth L1 error with respect to the ground-truth mesh. All the experiments are conducted on a desktop computer with and AMD EPYC 7542 CPU and NVIDIA RTX 3090 GPU.

For quantitative results, as shown in Table I of the poster, our method achieves superior rendering performance on the Replica dataset, improving PSNR by 0.89 dB and reducing LPIPS by 0.017 compared with the second-best method. Furthermore, as shown in Table II of the poster, our method achieves SOTA performance in mesh reconstruction on the Replica dataset, with Precision improved by 1.66, Recall by 0.24, and F1-score by 0.95, while the Depth L1 error is reduced by 0.02. As illustrated in Table III, the tracking error on the Replica dataset is reduced by 0.16 cm. As shown in Table IV, our method achieves superior rendering quality on the TUM RGB-D dataset, with PSNR improved by 0.9 over other advanced approaches. Moreover, the tracking accuracy on the TUM RGB-D dataset also exhibits competitive performance.

For qualitative results, as illustrated in Fig. 2 of the poster, our rendered images on the Replica dataset are clearer and exhibit fewer artifacts. Fig. 3 of the poster presents a visual comparison of meshes reconstructed by different methods on the Replica dataset, where our approach produces flatter surfaces and sharper edges. Fig. 4 and 5 show the 3D meshes reconstructed by our method on the TUM-RGBD dataset for the fr1/desk and fr2/xyz scenes. The results clearly demonstrate that noise is significantly smoothed, and the reconstructions faithfully capture the underlying geometric structures.