TCGSplat: Temporal Confidence Guided 3D Gaussian Splatting for RGB-D SLAM

Xingchen Guo
Changchun University of Science and Technology
Zhi Liu
liuzhi@cust.edu.cn
Changchun University of Science and Technology
Dandan Huang
Changchun University of Science and Technology
Xingzhao Wang
Changchun University of Science and Technology
Huiji Wang
Changchun University of Science and Technology
Huilin Jiang
Changchun University of Science and Technology

Research Article

Keywords: SLAM, 3D Gaussian Splatting, Temporal Confidence, Uncertainty Field

Posted Date: August 24th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-10579135/v1

Additional Declarations: No competing interests reported.

TCGSplat: Temporal Confidence Guided 3D Gaussian Splatting for RGB-D SLAM

Xingchen Guo[1,2], Zhi Liu[2,3*], Dandan Huang[3], Xingzhao Wang[2,3], Huiji Wang[2,3], Huilin Jiang[1,2]

[1]School of Optoelectronic Engineering, Changchun University of Science and Technology, Weixing Road, Changchun, 130022, Jilin, China.

[2*]Institute of Space Optoelectronic Technology, Changchun University of Science and Technology, Weixing Road, Changchun, 130022, Jilin, China.

[3]School of Electronic Information Engineering, Changchun University of Science and Technology, Weixing Road, Changchun, 130022, Jilin, China.

*Corresponding author(s). E-mail(s): liuzhi@cust.edu.cn;
Contributing authors: gxc_guxen@mails.cust.edu.cn; hdd@cust.edu.cn;

Abstract

3D Gaussian Splatting has been widely adopted in real-time dense Simultaneous Localization and Mapping systems due to its high-quality scene representation and efficient differentiable rendering. However, existing 3DGS-SLAM methods primarily optimize Gaussian primitives based on current observations, with limited explicit modeling of their time-varying reliability. Consequently, unreliable Gaussian primitives may adversely affect tracking, mapping, and loop closure. To address this issue, we propose Temporal Confidence Guided Splatting, a reliability-aware 3DGS-SLAM system. TCGSplat maintains a time-evolving confidence score for each Gaussian primitive to characterize its temporal reliability throughout online operation. The confidence score is adaptively updated using two observation cues: the mapping-induced geometric change of the Gaussian relative to its historical behavior and its rendering residual. Furthermore, TCGSplat renders the temporal confidence scores through Gaussian alpha compositing to generate a real-time confidence map, which guides camera tracking, map optimization, and loop closure viewpoint selection. Experiments on the synthetic Replica dataset and the real-world TUM RGB-D and ScanNet datasets demonstrate that TCGSplat improves system robustness and achieves competitive or superior performance compared with existing dense RGB-D SLAM methods in tracking accuracy, reconstruction quality, and rendering quality.The source code is available at https://github.com/gxcguxen/TCGSplat.

Keywords: SLAM, 3D Gaussian Splatting, Temporal Confidence, Uncertainty Field

1 Introduction

Simultaneous localization and mapping (SLAM) is a fundamental capability in robotics and augmented reality applications [1]. Classical dense SLAM methods typically rely on explicit geometric representations, such as voxels, TSDFs, surfels, or meshes, which support real-time tracking and mapping but have limited capability for high-fidelity appearance modeling and photorealistic rendering [2], [3]. Recently, neural radiance fields and other differentiable scene representations have been integrated into SLAM systems to jointly optimize camera poses and scene representations by minimizing differentiable rendering residuals [4–7]. However, implicit volumetric rendering is often computationally expensive, making it difficult to achieve both efficient online optimization and fine-grained scene reconstruction. 3D Gaussian Splatting (3DGS) represents a scene using anisotropic Gaussian primitives and enables efficient differentiable rendering through rasterization and alpha compositing [8]. It has therefore become an attractive scene representation for real-time RGB-D SLAM systems [9–16].

Existing 3DGS-SLAM methods primarily focus on the initialization, optimization, and rendering of Gaussian primitives, while camera poses and map parameters are commonly optimized using photometric and depth residuals. However, sensor noise, occlusions, viewpoint changes, and dynamic disturbances cause different image regions to provide constraints of varying reliability for camera tracking and map optimization. Recent studies have introduced uncertainty, visibility, and confidence modeling into neural fields, 3DGS, and SLAM systems to estimate the reliability of scene geometry, appearance, depth observations, and rendered predictions [17–19]. These methods generally use reliability estimates to reduce the influence of unreliable observations during optimization, thereby improving tracking and reconstruction robustness in challenging scenes. These studies suggest that an effective scene representation should encode not only geometry and appearance, but also reliability information that can be directly used by tracking and mapping.

Despite this progress, many existing approaches derive reliability estimates primarily from current observations or fixed attributes. In online mapping, individual Gaussians differ in how frequently they are observed, how strongly their geometric parameters are updated, and how much effective observation feedback they receive; their reliability therefore changes over time. To model this temporal evolution, we propose Temporal Confidence Guided Splatting (TCGSplat), a framework for temporal reliability modeling in RGB-D 3DGS-SLAM. TCGSplat maintains a time evolving confidence score for each Gaussian primitive and updates it online using temporal decay, rendering residuals, and comparisons between its current geometric change and historical behavior. TCGSplat renders the per Gaussian confidence scores into an image space confidence map through standard Gaussian alpha compositing. The resulting map emphasizes reliable regions during tracking, allocates more optimization effort to regions with low confidence during mapping, and reranks candidate viewpoints for submap registration. TCGSplat is implemented on top of an RGB-D 3DGS-SLAM system organized around local Gaussian submaps, while the backend follows LoopSplat [14] for submap retrieval, Gaussian map registration, and pose graph optimization.

The main contributions of this paper are summarized as follows:

(1) We propose a per Gaussian temporal confidence model for online 3DGS-SLAM that maintains an independent reliability state for each Gaussian using temporal decay, history aware geometric feedback, and RGB-D rendering residuals.

(2) We design a confidence guided optimization framework that renders per Gaussian confidence into image space weights through Gaussian alpha compositing. The resulting confidence map emphasizes reliable regions during tracking, allocates more optimization effort to low confidence regions during mapping, and supports viewpoint selection for loop closure.

(3) We conduct extensive experiments on the Replica, TUM RGB-D, and ScanNet datasets, demonstrating the effectiveness of TCGSplat in tracking accuracy, reconstruction quality, and rendering quality.

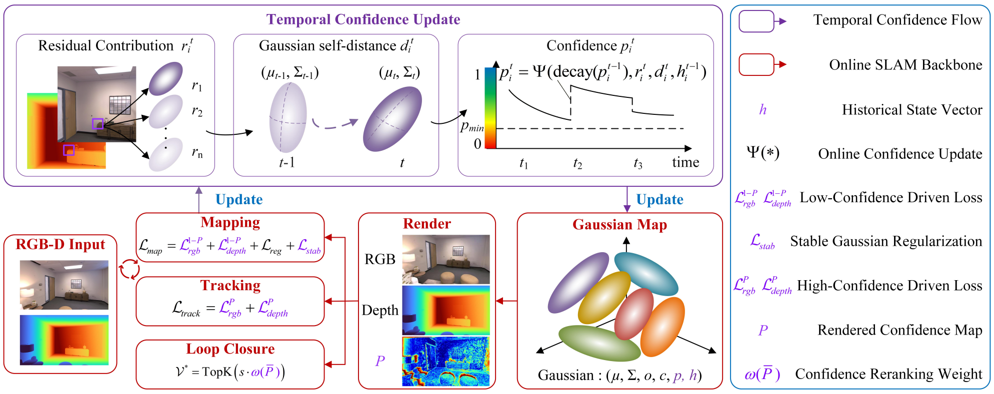

Fig. 1 Overview of TCGSplat. The system takes an RGB-D stream as input and maintains local Gaussian submaps for online tracking and mapping. The current submap is differentiably rendered into RGB, depth, and confidence maps. The confidence map is produced by alpha compositing the stored per-Gaussian confidence states, emphasizing reliable regions for tracking and increasing the optimization focus on low-confidence regions during mapping. After each mapping step, visible Gaussians provide observation feedback through rendering residuals and self-distance changes before and after optimization. Together with the historical state vector, the temporal confidence update function produces the new confidence state, which is written back to the Gaussian map and used by subsequent frames and submap-level view selection.

2 Related Work

2.1 Dense Visual SLAM

Traditional dense visual SLAM methods typically construct explicit geometric maps from RGB-D observations. KinectFusion [2] employs TSDF fusion for real-time camera tracking and surface reconstruction, while ElasticFusion [3] uses a surfel-based map to improve the flexibility and global consistency of dense mapping. Loop closure has also been widely studied to reduce accumulated drift in RGB-D SLAM, for example by combining global and local image features for more robust place recognition [20]. Although these methods achieve high geometric accuracy and real-time performance, they are less suited to high-fidelity appearance modeling and photorealistic rendering. With the development of neural scene representations, iMAP [21] pioneered the joint formulation of scene representation and camera tracking using an online-optimized implicit neural field. NICE-SLAM [5], ESLAM [6], and Point-SLAM [7] subsequently improved scalability, efficiency, and reconstruction detail through local feature encodings, hybrid representations, or neural point clouds. Despite their enhanced scene representation capability, neural implicit methods still incur considerable computational costs due to volumetric rendering and continual network optimization.

2.2 3DGS-based SLAM

3D Gaussian Splatting [8] represents a scene using explicit Gaussian primitives and enables high-quality real-time rendering through differentiable rasterization and alpha compositing, providing online SLAM with an efficient and expressive map representation. GS-SLAM [9] combines incremental Gaussian expansion with coarse-to-fine pose optimization, SplaTAM [10] establishes an online tracking and mapping pipeline tailored to the Gaussian representation, and Gaussian-SLAM [11] employs submaps to improve online optimization efficiency in large scenes. To address accumulated drift during long-term operation, LoopSplat [14] further maintains global consistency through Gaussian submap registration and pose graph optimization. Recent studies have also investigated robust 3DGS training and rendering under low-quality observations [22], showing that the quality and reliability of input evidence can strongly affect Gaussian scene representations. Although these representative methods substantially improve tracking, mapping, and rendering performance, the Gaussian updates in many existing systems are primarily driven by current observation residuals, without explicitly modeling how the reliability of each primitive evolves with its observation history.

2.3 Uncertainty-aware Field SLAM

Uncertainty estimation has been explored in both radiance fields and dense SLAM. Stochastic NeRF [23] employs Bayesian variational inference to estimate predictive uncertainty, while Bayes' Rays [24] and FisherRF [25] quantify spatial or model uncertainty for reconstruction and informative view selection. View selection has also been introduced into point-based NeRF rendering to improve low-quality regions by selecting more informative observations [26]. In online neural SLAM, UncLe-SLAM [27] and Uni-SLAM [28] use depth or predictive uncertainty to reweight tracking and mapping losses. For 3DGS-SLAM, CG-SLAM [17] models uncertainty from depth and Gaussian geometry consistency, whereas VarSplat [18] learns per-Gaussian appearance variance and renders a per-pixel uncertainty map through alpha compositing. These methods primarily characterize uncertainty in input depth, geometry, or appearance, but give less attention to how the reliability of an individual Gaussian evolves through repeated observation, continual optimization, and prolonged absence of observation. In contrast, TCGSplat maintains a time evolving confidence state for each Gaussian and updates it using rendering residuals and geometric change relative to a historical exponential moving average (EMA) baseline. The resulting confidence provides unified reliability guidance for tracking, mapping, and viewpoint selection for submap registration.

3 Methods

3.1 System Overview

TCGSplat is built upon a submap-based 3DGS-SLAM framework that takes an RGB-D image sequence as input. Given an input frame $\mathcal{F}_t = \{\mathbf{I}_t, \mathbf{D}_t\}$ at time t, the system alternates between camera tracking and local mapping. Tracking estimates the camera pose $\mathbf{T}_t$ by minimizing the color and depth residuals between the input observation and the rendering of the current Gaussian submap. Mapping initializes Gaussian primitives in newly observed regions and jointly optimizes their geometric and appearance parameters. When the camera motion satisfies the submap-switching criterion, the current submap is saved and a new local Gaussian map is created. For completed submaps, the system follows the LoopSplat [14] backend for submap retrieval, Gaussian map registration, and pose graph optimization to correct accumulated pose drift and maintain global consistency. As illustrated in Fig. 1, TCGSplat augments this pipeline with a temporal confidence update module, which maintains per-Gaussian reliability states from observation feedback and renders them as confidence maps to guide tracking, mapping, and submap-level processing.

Each submap is represented by a set of anisotropic Gaussian primitives $\mathcal{G} = \{G_i\}_{i=1}^{N}$. The basic parameters of the i-th Gaussian are defined as

$$G_i = (\boldsymbol{\mu}_i, \mathbf{s}_i, \mathbf{q}_i, o_i, \mathbf{c}_i)\,, \tag{1}$$

where $\boldsymbol{\mu}_i \in \mathbb{R}^3$ denotes its 3D center, $\mathbf{s}_i \in \mathbb{R}^3$ and the unit quaternion $\mathbf{q}_i \in \mathbb{R}^4$ describe its scale and rotation, respectively, o_i is the opacity, and $\mathbf{c}_i$ denotes the view-dependent color encoded by spherical harmonics. The 3D covariance matrix is constructed from the scale and rotation as

$$\boldsymbol{\Sigma}_i = \mathbf{R}(\mathbf{q}_i)\,\mathrm{diag}(\mathbf{s}_i^2)\mathbf{R}(\mathbf{q}_i)^{\mathsf{T}} \tag{2}$$

where $\mathbf{R}(\mathbf{q}_i)$ denotes the rotation matrix corresponding to the quaternion $\mathbf{q}_i$. Under a given camera pose and projection model, the 3D Gaussian is projected onto the image plane as a 2D Gaussian with mean $\boldsymbol{\mu}_i'$ and covariance $\boldsymbol{\Sigma}_i'$. Its alpha contribution at pixel $\mathbf{u}$ is then defined as

$$\alpha_i(\mathbf{u}) = o_i \exp\left[-\frac{1}{2}(\mathbf{u}-\boldsymbol{\mu}'_i)^\top (\boldsymbol{\Sigma}'_i)^{-1}(\mathbf{u}-\boldsymbol{\mu}'_i)\right]. \tag{3}$$

During rendering, the visible Gaussians are sorted by depth and composited from front to back according to their pixel level alpha contributions and accumulated transmittance. This process produces the rendered RGB image and depth map while accounting for color, depth, opacity, and occlusion. Since the rendering operation is differentiable with respect to both camera poses and Gaussian parameters, it provides the gradients required for tracking and mapping optimization.

In addition to the standard Gaussian parameters, TCGSplat maintains a temporal confidence state $p_i^t \in [p_{\min}, 1]$ and a historical state vector $\mathbf{h}_i^t$ for each Gaussian primitive. At frame t, tracking and mapping use the confidence state p_i^{t-1} obtained from the previous frame. After mapping, observation feedback from visible Gaussians is used to update p_i^{t-1} to p_i^t, which is stored for subsequent frames. The confidence states are rendered through Gaussian alpha compositing to produce an image space confidence map for tracking, mapping, and submap registration. The detailed update and guidance mechanisms are described in the following sections.

3.2 Per-Gaussian Temporal Confidence Modeling

Gaussian primitives constructed during online mapping have different creation times and observation histories, causing their reliability to evolve with continued observation and optimization. TCGSplat therefore maintains a confidence state p_i^t and a historical state vector $\mathbf{h}_i^t = (\bar{d}_i^t, b_i, n_{i,\mathrm{EMA}}^t)$ for each Gaussian G_i. Here, $\bar{d}_i^t$ denotes the exponential moving average (EMA) of its self-distance, b_i denotes its creation time within the current submap, and $n_{i,\mathrm{EMA}}^t$ records the number of valid EMA updates up to time t. These states are updated directly from online observations rather than optimized as learnable parameters through backpropagation.

After the mapping step at frame t, the confidence of each Gaussian first undergoes bounded decay toward the lower bound $p_{\min}$:

$$p_{i,-}^t = p_{\min} + \left(p_i^{t-1} - p_{\min}\right)\exp\left(-\frac{\Delta t}{\tau}\right), \tag{4}$$

where $p_{i,-}^t$ denotes the decayed intermediate confidence before incorporating the current observation feedback, and τ is the decay time constant. The current online implementation uses $\Delta t = 1$. Gaussians with valid observation feedback are subsequently reinforced or penalized, whereas those without valid feedback retain only the decayed confidence. The lower bound $p_{\min}$ preserves their basic contribution and prevents map elements from becoming completely inactive.

Throughout this section, the map state obtained after the mapping step of frame $t-1$ is used as the initial map state of frame t. Therefore, for a Gaussian that exists before the current mapping step, $(\boldsymbol{\mu}_i^{t-1}, \mathbf{s}_i^{t-1}, \mathbf{q}_i^{t-1})$ and $(\boldsymbol{\mu}_i^t, \mathbf{s}_i^t, \mathbf{q}_i^t)$ denote its geometric parameters immediately before and after the mapping update at frame t, respectively.

For a Gaussian that is visible in the current view and can be consistently associated before and after mapping, its component-wise geometric changes are defined as

$$d_{\mu,i}^t = \left\| \boldsymbol{\mu}_i^t - \boldsymbol{\mu}_i^{t-1} \right\|_2, \tag{5}$$

$$d_{s,i}^t = \left\| \log\left(\mathbf{s}_i^t + \epsilon\right) - \log\left(\mathbf{s}_i^{t-1} + \epsilon\right) \right\|_2, \tag{6}$$

$$d_{q,i}^t = 1 - \left| \left\langle \mathbf{q}_i^t, \mathbf{q}_i^{t-1} \right\rangle \right|^2. \tag{7}$$

The Gaussian self-distance is then defined as

$$d_i^t = d_{\mu,i}^t + \lambda_s d_{s,i}^t + \lambda_q d_{q,i}^t, \tag{8}$$

where λ_s and λ_q control the contributions of the scale and rotation changes, respectively, and ϵ is a small positive constant for numerical stability. The logarithm in the scale term is applied elementwise. The quaternions are normalized before comparison, and the absolute inner product makes the rotation term invariant to the equivalent quaternion representations $\mathbf{q}$ and $-\mathbf{q}$. In the implementation, a unique Gaussian ID establishes the correspondence of each primitive before and after mapping, preventing index changes caused by densification or pruning from corrupting the self-distance computation.

In addition to geometric change, the rendering residual measures how well a Gaussian explains the current observation. We construct a combined RGB-D residual map as

$$R_t(\mathbf{u}) = \lambda_{\text{rgb}} r_{\text{rgb}}^t(\mathbf{u}) + \lambda_d M_D^t(\mathbf{u}) r_d^t(\mathbf{u}), \tag{9}$$

where $r_{\text{rgb}}^t(\mathbf{u})$ and $r_d^t(\mathbf{u})$ denote the pixel level color and depth residuals, respectively, and $M_D^t(\mathbf{u})$ is the valid depth mask. The residual associated with Gaussian G_i is obtained by sampling R_t at its projected center $\mathbf{u}_i^t$, yielding $r_i^t = R_t(\mathbf{u}_i^t)$. The sampled residual is converted into an observation quality score as

$$Q_{r,i}^t = \exp\left(-\frac{r_i^t}{r_{\text{ref}}}\right). \tag{10}$$

To account for the intrinsic change scale of each Gaussian, TCGSplat does not evaluate d_i^t using a uniform threshold. Instead, the current self-distance is compared with the Gaussian specific EMA baseline $\bar{d}_i^{t-1}$ before the baseline is updated:

$$\xi_i^t = \tanh\left(\frac{\bar{d}_i^{t-1} - d_i^t}{\sigma_d}\right), \tag{11}$$

where σ_d controls the sensitivity to deviations from the historical baseline. When $d_i^t < \bar{d}_i^{t-1}$, $\xi_i^t > 0$, indicating that the Gaussian changes less than expected from its historical behavior. Conversely, $\xi_i^t < 0$ indicates that the current mapping update produces a larger than expected geometric change.

Based on this relative stability signal, the positive excitation and negative penalty are defined as

$$A_i^t = \gamma_+ Q_{r,i}^t [\xi_i^t]_+, \qquad B_i^t = \gamma_- [-\xi_i^t]_+, \tag{12}$$

where $[x]_+ = \max(x, 0)$, and γ_+ and γ_- control the strengths of the positive and negative feedback, respectively. The observation quality score $Q^t_{r,i}$ modulates only the positive excitation. Therefore, a Gaussian with a small geometric change but a large rendering residual is not strongly reinforced, whereas an unusually large self-distance directly produces a negative penalty.

The confidence is then updated by combining the decayed state with the positive excitation and negative penalty:

$$p^t_i = \mathrm{clip}_{[p_{\min},1]}\left(p^t_{i,-} + \eta A^t_i \left(1 - p^t_{i,-}\right) - \eta B^t_i \left(p^t_{i,-} - p_{\min}\right)\right), \tag{13}$$

where η is the confidence update rate. The multiplicative boundary terms reduce the positive update as the confidence approaches 1 and reduce the negative update as it approaches $p_{\min}$.

When $n^{t-1}_{i,\mathrm{EMA}} = 0$, Gaussian G_i has no historical baseline for relative comparison. Its first positive excitation is therefore initialized using the absolute self-distance and rendering residual:

$$A^t_i = \exp\left(-\frac{d^t_i}{d_{\mathrm{ref}}} - \frac{r^t_i}{r_{\mathrm{ref}}}\right), \qquad B^t_i = 0, \tag{14}$$

where d_{ref} and r_{ref} are reference scales for the self-distance and rendering residual, respectively. The EMA baseline is initialized as $\bar{d}^t_i = d^t_i$, and the number of valid EMA updates is incremented accordingly.

For subsequent valid observations, the EMA baseline is updated only after the confidence update:

$$\bar{d}^t_i = (1 - \beta)\bar{d}^{t-1}_i + \beta d^t_i, \tag{15}$$

where β controls the contribution of the current self-distance to the historical baseline. The corresponding EMA update counter $n^t_{i,\mathrm{EMA}}$ is incremented after each valid update.

Gaussians within the same submap may be created at different times. Consequently, Gaussians created later undergo fewer decay updates before the submap is completed. Let b_i denote the number of mapping updates that occurred before Gaussian G_i was created. Before saving the completed submap, TCGSplat compensates for the b_i decay updates missed before its creation:

$$p_{i,\mathrm{sub}} = p_{\min} + \left(p^t_i - p_{\min}\right)\exp\left(-\frac{b_i}{\tau}\right), \tag{16}$$

where p^t_i denotes the confidence state immediately before the current submap is saved. This calibration places Gaussians created at different times on a comparable temporal scale and prevents recently created primitives from receiving an artificial confidence advantage solely because they have undergone fewer decay updates.

3.3 Confidence-guided Tracking, Mapping, and Loop Closure

At frame t, all confidence guided operations use the states p^{t-1}_i obtained after the previous mapping step. The updated states p^t_i are used from frame $t+1$ onward.

To transfer the per Gaussian confidence states into image space, TCGSplat treats p_i^{t-1} as a scalar Gaussian attribute and renders it using the same depth order and alpha compositing process as RGB and depth:

$$P_t(\mathbf{u}) = \sum_{i \in \mathcal{V}_t(\mathbf{u})} T_i(\mathbf{u}) \alpha_i(\mathbf{u}) p_i^{t-1}, \tag{17}$$

where $T_i(\mathbf{u}) = \prod_{j<i}(1 - \alpha_j(\mathbf{u}))$ is the accumulated transmittance before Gaussian G_i, and $\mathcal{V}_t(\mathbf{u})$ denotes the depth ordered Gaussians contributing to pixel $\mathbf{u}$. The resulting confidence map P_t is used for tracking and mapping and is also rendered at candidate viewpoints during submap registration.

During tracking, the Gaussian map is fixed and only the current camera pose is optimized. The rendered confidence map is used to construct the pixel wise tracking weight

$$W_{\mathrm{tr}}^t(\mathbf{u}) = w_{\min} + (1 - w_{\min})\, P_t(\mathbf{u}), \tag{18}$$

which is applied to the color and depth residuals in the baseline tracking objective. The tracking loss can be written compactly as

$$\mathcal{L}_{\mathrm{tr}} = \sum_{\mathbf{u}} W_{\mathrm{tr}}^t(\mathbf{u}) \left[\lambda_c \mathcal{L}_{\mathrm{rgb}}^t(\mathbf{u}) + \lambda_d \mathcal{L}_{\mathrm{depth}}^t(\mathbf{u})\right], \tag{19}$$

where $\mathcal{L}_{\mathrm{rgb}}^t$ and $\mathcal{L}_{\mathrm{depth}}^t$ denote the color and depth residual terms used by the baseline system. Regions supported by high confidence Gaussians therefore provide stronger constraints for pose estimation, while $w_{\min} > 0$ preserves basic constraints from regions with low confidence.

During mapping, regions with low confidence are assigned more optimization effort. We therefore define the reconstruction weight as

$$W_{\mathrm{map}}^t(\mathbf{u}) = 1 + (1 - P_t(\mathbf{u})), \tag{20}$$

which is applied to the color and depth reconstruction terms in the baseline mapping objective. The mapping objective can be written compactly as

$$\mathcal{L}_{\mathrm{map}} = \sum_{\mathbf{u}} W_{\mathrm{map}}^t(\mathbf{u}) \left[\lambda_c \mathcal{L}_{\mathrm{rgb}}^t(\mathbf{u}) + \lambda_d \mathcal{L}_{\mathrm{depth}}^t(\mathbf{u})\right] + \lambda_s s \mathcal{L}_{\mathrm{ssim}} + \lambda_r \mathcal{L}_{\mathrm{reg}} + \mathcal{L}_{\mathrm{stab}}, \tag{21}$$

where $\mathcal{L}_{\mathrm{ssim}}$ and $\mathcal{L}_{\mathrm{reg}}$ follow the baseline mapping objective, and $\mathcal{L}_{\mathrm{stab}}$ is the adaptive stability constraint introduced below. Regions with low confidence receive larger reconstruction weights, whereas regions with high confidence retain unit weight.

To suppress abrupt geometry changes of mature and reliable Gaussians, TCGSplat further introduces a lightweight stability loss based on the self-distance defined in Eq. (8). Instead of constraining the absolute deviation from a fixed reference state, we penalize only the abnormal increase of the current self-distance over its historical EMA baseline:

$$\Delta d_i^t = \left[d_i^t - \bar{d}_i^{t-1}\right]_+. \tag{22}$$

Let $a_i^t = \max(t - b_i, 0)$ denote the age of Gaussian i within the current submap. The maturity-aware stability score is defined as $\kappa_i^t = p_i^{t-1}(1 - \exp(-a_i^t/\tau_a))$. The stability loss is then

$$\mathcal{L}_{\text{stab}} = \lambda_{\text{stab}} \frac{1}{|\mathcal{V}|} \sum_{i \in \mathcal{V}} \left[\Delta d_i^t - \left(\delta_{\max} - \kappa_i^t \left(\delta_{\max} - \delta_{\min} \right) \right) \right]_+ , \tag{23}$$

where $\mathcal{V}$ is the set of currently visible Gaussians. As confidence and age increase, the allowed abnormal change becomes smaller, while newly created or low-confidence Gaussians retain greater flexibility.

In addition to image space reweighting and the stability constraint, temporal confidence controls the optimization strength of Gaussian geometry. For Gaussian G_i, the gradient scaling factor is defined as

$$g_i^t = g_{\min} + (g_{\max} - g_{\min}) \frac{1 - p_i^{t-1}}{1 - p_{\min}}, \tag{24}$$

where $g_{\min}$ and $g_{\max}$ denote the minimum and maximum gradient scaling factors, respectively. The normalization ensures that $p_i^{t-1} = 1$ corresponds to $g_i^t = g_{\min}$, whereas $p_i^{t-1} = p_{\min}$ corresponds to $g_i^t = g_{\max}$. The factor g_i^t is applied only to the gradients of the Gaussian center, scale, and rotation during backpropagation. Consequently, Gaussians with lower confidence receive stronger geometric updates, whereas Gaussians with higher confidence are optimized more conservatively.

During loop closure, TCGSplat retains the submap retrieval, Gaussian registration, and pose graph optimization pipeline of LoopSplat [14]. Temporal confidence is used only to rerank the candidate registration viewpoints. For each candidate viewpoint v, we compute its mean confidence C_v over the high-opacity region and adjust the original descriptor score as

$$\widetilde{S}_v = S_v \left[w_{\min}^{\text{lc}} + \left(1 - w_{\min}^{\text{lc}} \right) C_v \right] . \tag{25}$$

The reranking is performed only among viewpoints preselected by descriptor similarity, preserving the original retrieval process while reducing the use of views dominated by unreliable map regions.

4 Experiments

4.1 Experimental Setup

We evaluate TCGSplat on three public datasets: Replica [29], TUM RGB-D [30], and ScanNet [31]. Replica is a synthetic RGB-D dataset generated from high-quality indoor scene models and provides accurate camera trajectories and geometric ground truth, enabling a comprehensive evaluation of pose estimation, rendering quality, and scene reconstruction. TUM RGB-D consists of real indoor sequences captured with a handheld RGB-D camera and includes sensor noise, motion blur, and rapid viewpoint changes. ScanNet contains larger-scale real indoor scenes with more complex geometry and appearance. All experiments are conducted on Ubuntu 22.04 using an NVIDIA GeForce RTX 4090 GPU.

TCGSplat is implemented upon the submap-based RGB-D 3DGS-SLAM framework of LoopSplat [14]. Except for the proposed temporal confidence module, the keyframe management, Gaussian map registration, and pose graph optimization remain consistent with the base system. Replica and TUM RGB-D adopt a motion-triggered submap partitioning strategy: the current submap is terminated and a new one is created when the camera rotation or translation relative to the starting pose of the submap exceeds 50° or $0.5\,\mathrm{m}$, respectively. ScanNet instead uses a fixed interval and creates a new submap every 50 input frames. The remaining tracking, mapping, and loop-closure parameters follow the settings of LoopSplat [14].

For confidence guidance, the minimum pixel weight in tracking is set to 0.2 to preserve basic constraints from low-confidence regions. The geometric gradient scaling range in mapping is set to $[0.8, 1.2]$, and the weight of the adaptive stability loss is set to 0.01. During loop closure, the six viewpoints with the highest descriptor scores are first preselected, from which two viewpoints are selected for subsequent registration by incorporating their mean view confidence. For TCGSplat and the reproduced baselines, each sequence is evaluated once using a fixed random seed and the same evaluation scripts.

4.2 Baselines and Evaluation Metrics

We compare TCGSplat with representative neural implicit SLAM and 3DGS-SLAM methods. For neural implicit representations, we include NICE-SLAM [5], GO-SLAM [32], and Loopy-SLAM [15], covering classical implicit scene representations and dense SLAM systems with global optimization or loop-closure constraints. For methods based on 3D Gaussian Splatting, we compare against SplaTAM [10], Gaussian-SLAM [11], LoopSplat [14], CG-SLAM [17], and VarSplat [18], covering representative Gaussian SLAM systems with online tracking and mapping, submap-based reconstruction, global consistency optimization, and reliability-related scene modeling. For each baseline, we prioritize publicly reported results that have been adopted by multiple related works under the same evaluation protocol. When reproducible official implementations are available, we rerun them under the corresponding settings. Methods marked with * in the tables are reproduced using official implementations.

For tracking, we evaluate camera trajectories using the root mean square error of the absolute trajectory error (ATE RMSE), reported in centimeters after rigid alignment between the estimated and ground-truth trajectories. For reconstruction quality, we evaluate Replica scenes with geometric ground truth using Depth L1, the mean absolute error between the rendered and observed depth maps, and F1 score to measure surface accuracy and completeness. For rendering quality, we report the peak signal-to-noise ratio (PSNR), structural similarity index measure (SSIM), and learned perceptual image patch similarity (LPIPS). Higher PSNR, SSIM, and F1 score indicate better rendering or reconstruction quality, whereas lower ATE RMSE, LPIPS, and Depth L1 indicate better trajectory accuracy, perceptual quality, and geometric consistency. The results are highlighted as **best**, second and third.

4.3 Quantitative Evaluation

4.3.1 Tracking

Tables 1, 2, and 3 compare the tracking accuracy on Replica, ScanNet, and TUM RGB-D using ATE RMSE in centimeters. TCGSplat achieves competitive or superior results across all three datasets. On Replica, it reaches an average ATE of 0.25 cm, matching VarSplat and slightly outperforming LoopSplat. On ScanNet, TCGSplat obtains the best average result among 3DGS-based methods, reducing the ATE from 8.5 cm for LoopSplat and 7.9 cm for VarSplat to 7.6 cm. On TUM RGB-D, TCGSplat achieves the lowest average ATE of 3.37 cm among the compared 3DGS-SLAM methods. These results suggest that temporal confidence provides useful reliability cues for pose optimization. By emphasizing regions supported by stable Gaussians, TCGSplat reduces the influence of unreliable map primitives, especially in real-world sequences with sensor noise, occlusions, and challenging camera motion. Fig. 2 further provides a qualitative comparison on ScanNet scenes 0000, 0059, and 0181, where trajectories are overlaid on top-down mesh views.

Table 1 Tracking performance on Replica (ATE RMSE↓ [cm]).

Method	R0	R1	R2	Off0	Off1	Off2	Off3	Off4	Avg.
Neural Fields									
NICE-SLAM	0.97	1.31	1.07	0.88	1.00	1.06	1.10	1.13	1.06
GO-SLAM	0.34	0.29	0.29	0.32	0.30	0.39	0.39	0.46	0.35
Loopy-SLAM*	0.45	**0.23**	0.31	0.25	0.21	0.33	0.30	0.35	0.30
3DGS									
SplaTAM	0.31	0.40	0.29	0.47	0.27	0.29	0.32	0.72	0.38
Gaussian-SLAM*	0.31	0.26	0.22	0.29	0.19	0.33	0.39	0.40	0.30
LoopSplat*	0.25	0.24	0.20	**0.23**	0.20	0.37	0.19	0.36	0.26
CG-SLAM	0.29	0.27	0.25	0.33	**0.14**	**0.28**	0.31	**0.29**	0.27
VarSplat*	**0.20**	0.25	0.25	0.26	0.15	0.35	0.19	0.35	**0.25**
TCGSplat(Ours)	0.26	0.24	**0.14**	0.24	0.19	0.36	**0.18**	0.37	**0.25**

4.3.2 Reconstruction

Table 4 reports the reconstruction performance on Replica using Depth L1 and F1 score. TCGSplat achieves an average Depth L1 of 0.85 and an average F1 score of 90.2%, showing competitive reconstruction quality among both neural-field and 3DGS-based methods. Compared with the submap-based 3DGS baselines, TCGSplat reduces the average Depth L1 from 0.88 for LoopSplat and 0.89 for VarSplat to 0.85, while maintaining the same average F1 score. The improvement is especially visible on challenging scenes such as Office3, where TCGSplat produces lower depth error than LoopSplat and VarSplat. These results suggest that confidence guided mapping improves depth consistency without degrading surface completeness. By assigning

Table 2 Tracking performance on ScanNet (ATE RMSE↓ [cm]).

Method	0000	0059	0106	0169	0181	0207	Avg.
Neural Fields							
NICE-SLAM	12.0	14.0	7.9	10.9	13.4	6.2	10.7
GO-SLAM	5.4	7.5	**7.0**	7.0	6.8	6.9	**6.8**
Loopy-SLAM*	5.4	7.8	8.7	9.5	10.6	7.9	8.3
3DGS							
SplaTAM	10.1	17.7	11.7	7.5	**5.6**	7.5	10.0
Gaussian-SLAM*	21.2	12.8	13.5	13.6	21.0	13.4	15.9
LoopSplat*	5.6	6.3	8.3	9.4	14.1	7.4	8.5
CG-SLAM	7.1	7.5	8.9	8.2	11.6	**5.3**	8.1
VarSplat*	5.1	5.4	8.3	**6.6**	15.3	6.3	7.9
TCGSplat(Ours)	**4.9**	**5.0**	7.1	10.6	10.4	7.5	7.6

Table 3 Tracking performance on TUM RGB-D (ATE RMSE ↓ [cm]).

Method	fr1/desk	fr1/desk2	fr1/room	fr2/xyz	fr3/off.	Avg.
Neural Fields						
NICE-SLAM	4.26	4.99	34.49	6.19	3.87	10.76
GO-SLAM	**1.50**	N/A	**4.64**	**0.60**	**1.30**	N/A
Loopy-SLAM*	3.96	**3.03**	8.99	1.90	3.31	4.24
3DGS						
SplaTAM	3.35	6.54	11.13	1.24	5.16	5.48
Gaussian-SLAM*	2.79	6.21	12.82	1.62	6.90	6.07
LoopSplat*	2.02	3.62	6.49	1.55	3.40	3.42
CG-SLAM	2.43	4.54	9.39	1.20	2.45	4.00
VarSplat*	2.03	3.59	6.47	1.44	3.71	3.45
TCGSplat(Ours)	1.83	3.52	6.74	1.45	3.31	**3.37**

larger reconstruction weights and stronger geometric updates to regions with low confidence, TCGSplat encourages further refinement of insufficiently optimized Gaussian primitives while applying conservative updates to mature and reliable structures.

4.3.3 Rendering

Table 5 compares the rendering quality on Replica, TUM RGB-D, and ScanNet using PSNR, SSIM, and LPIPS. TCGSplat achieves the highest PSNR on all three datasets, reaching 37.84 dB on Replica, 22.83 dB on TUM RGB-D, and 24.95 dB on ScanNet. On Replica, it matches the best SSIM result and improves LPIPS over the submap based baselines LoopSplat and VarSplat. On TUM RGB-D, TCGSplat matches VarSplat in SSIM while slightly improving its PSNR and LPIPS. On ScanNet, it consistently outperforms LoopSplat and VarSplat across all three rendering metrics. Although SplaTAM remains stronger on some SSIM and LPIPS comparisons, TCGSplat achieves the most consistent PSNR performance across the three datasets. These results indicate that temporal confidence benefits not only camera tracking but also the optimized

Table 4 Reconstruction performance on Replica

Method	Metric	R0	R1	R2	Off0	Off1	Off2	Off3	Off4	Avg.
Neural Fields										
NICE-SLAM	Depth L1↓	1.81	1.44	2.04	1.39	1.76	8.33	4.99	2.01	2.97
	F1[%]↑	45.0	44.8	43.6	50.0	51.9	39.2	39.9	36.5	43.9
Loopy-SLAM*	Depth L1↓	0.50	0.37	**0.37**	**0.32**	0.25	2.38	**0.51**	0.47	**0.65**
	F1[%]↑	**91.5**	**92.4**	90.4	**93.9**	**91.6**	88.4	**89.0**	**88.6**	**90.7**
3DGS										
SplaTAM	Depth L1↓	**0.43**	0.38	0.54	0.44	0.66	**1.05**	1.60	0.68	0.72
	F1[%]↑	89.3	88.2	88.0	91.7	90.0	85.1	77.1	80.1	86.2
Gaussian-SLAM*	Depth L1↓	0.83	0.63	0.70	0.55	**0.20**	2.94	3.47	0.70	1.25
	F1[%]↑	88.9	91.4	90.5	92.5	90.1	87.5	84.3	87.4	89.1
LoopSplat*	Depth L1↓	0.45	0.28	0.39	0.46	0.21	1.22	3.67	**0.35**	0.88
	F1[%]↑	90.7	91.9	91.1	93.3	90.3	88.7	87.9	88.2	90.2
VarSplat*	Depth L1↓	0.44	**0.27**	0.40	0.45	**0.20**	1.25	3.74	**0.35**	0.89
	F1[%]↑	90.6	91.8	91.0	93.3	90.3	**88.8**	87.6	88.2	90.2
TCGSplat(Ours)	Depth L1↓	0.45	0.28	0.39	0.44	0.21	1.27	3.38	0.36	0.85
	F1[%]↑	90.6	91.9	**91.2**	93.3	90.3	**88.8**	87.7	88.2	90.2

Gaussian representation, leading to improved rendering quality over the underlying submap based baseline.

Table 5 Rendering performance on Three Datasets

Dataset	Replica			TUM			ScanNet		
Method	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
NICE-SLAM	24.42	0.892	0.233	14.86	0.614	0.441	17.54	0.621	0.548
Loopy-SLAM*	32.18	0.933	0.184	19.26	0.663	0.521	15.44	0.601	0.674
SplaTAM	34.11	0.970	**0.100**	22.80	**0.893**	**0.178**	19.14	0.716	**0.358**
Gaussian-SLAM*	34.46	0.974	0.139	22.56	0.841	0.293	20.54	0.711	0.532
LoopSplat*	37.11	0.986	0.110	22.20	0.842	0.323	24.85	0.846	0.408
VarSplat*	37.22	**0.987**	0.108	22.80	0.867	0.281	24.83	0.845	0.407
TCGSplat(Ours)	**37.84**	**0.987**	0.107	**22.83**	0.867	0.279	**24.95**	**0.849**	0.402

4.4 Ablation and Analysis

Table 6 evaluates the cumulative contribution of the confidence guided components on ScanNet scene 0181. Starting from the LoopSplat baseline, confidence guided mapping reduces the ATE RMSE from 14.1 cm to 12.8 cm. Adding confidence guided tracking further reduces the error to 12.1 cm, while incorporating loop viewpoint reranking achieves the best result of 10.4 cm. On this challenging sequence, the results indicate that the three components provide complementary improvements to pose estimation.

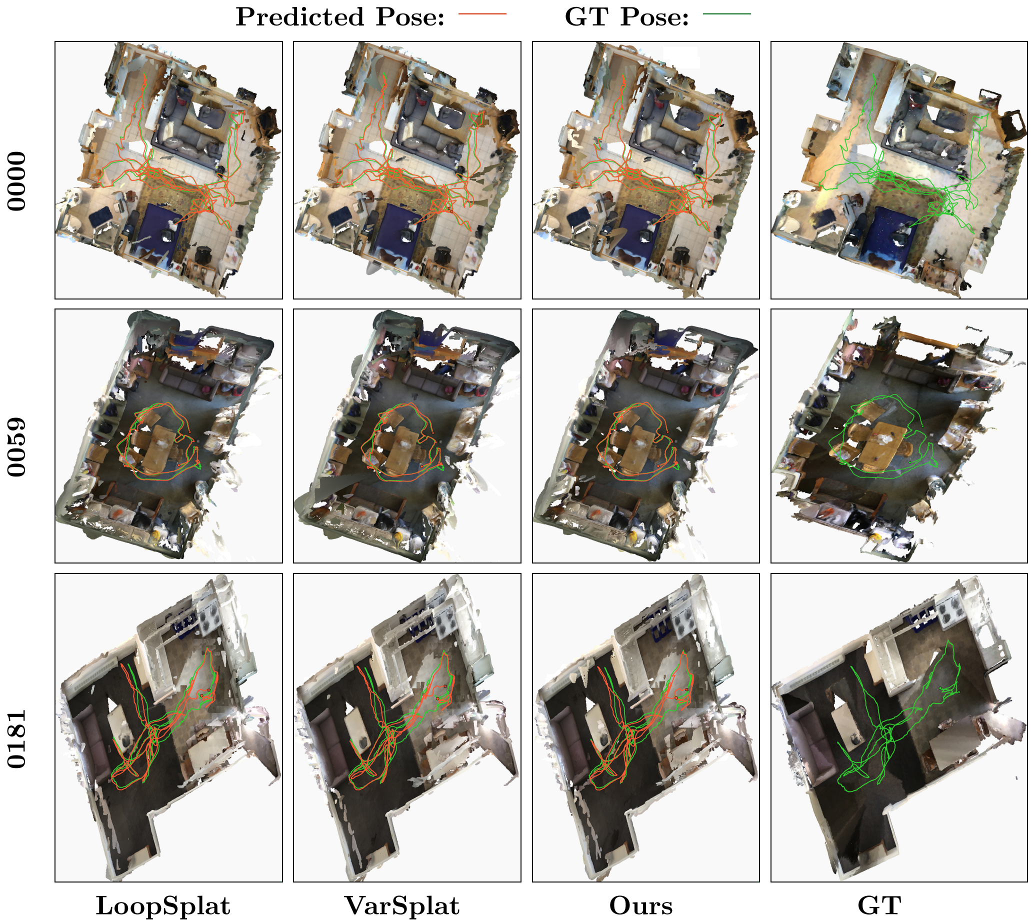

Fig. 2 Qualitative trajectory comparison on ScanNet scenes. Rows correspond to scenes 0000, 0059, and 0181, while columns show LoopSplat, VarSplat, TCGSplat, and ground-truth trajectories rendered on top-down mesh views.

Fig. 3 visualizes the temporal behavior of the confidence map on ScanNet scenes 0000 and 0059. For each scene, the first row shows input frames, the second row shows online rendered images, and the third row shows the corresponding confidence maps at different timestamps. The white boxes highlight regions that are well reconstructed in the rendered images; their corresponding confidence responses become more reliable as the scene is repeatedly observed. In contrast, the red boxes indicate newly revealed regions caused by viewpoint changes. These regions are not yet sufficiently optimized in the online rendering, and the confidence maps assign them lower confidence according to the visualization colormap. This qualitative result shows that temporal confidence evolves consistently with observation history and reconstruction quality, providing meaningful guidance for subsequent tracking and mapping.

Table 7 reports the runtime on Replica/Room0 using an RTX 4090 GPU. TCGSplat introduces additional confidence rendering, update, and guidance operations, but the overall overhead remains moderate. Compared with LoopSplat, TCGSplat increases the mapping iteration time from 11.28 ms to 14.82 ms and the tracking iteration time from 11.20 ms to 12.23 ms. Meanwhile, it remains more efficient per iteration than VarSplat in both mapping and tracking. The mapping and tracking time per frame are 0.26 s and 1.31 s, respectively, showing that TCGSplat improves robustness and rendering quality while maintaining practical runtime efficiency for online RGB-D SLAM.

Table 6 Ablation study of the TCG model on pose estimation. (ATE RMSE ↓ [cm]). Mapping includes confidence guided reconstruction weighting, geometric gradient scaling, and the adaptive stability constraint; Tracking denotes confidence weighted pose optimization; and Loop denotes confidence based viewpoint reranking.

Method	Mapping	Tracking	Loop	ATE RMSE↓
I	✗	✗	✗	14.1
II	✓	✗	✗	12.8
III	✓	✓	✗	12.1
IV	✓	✓	✓	**10.4**

Table 7 Runtime on Replica/Room0 using RTX4090 24GB.

Method	Mapping /fr(s)	Mapping iter(ms)	Tracking /fr(s)	Tracking iter(ms)
Gaussian-SLAM	0.34	11.35	1.50	11.21
LoopSplat	0.34	**11.28**	1.49	**11.20**
VarSplat	**0.20**	19.64	**1.01**	16.51
TCGSplat	0.26	14.82	1.31	12.23

5 Conclusion

In this paper, we presented TCGSplat, a temporal confidence guided 3DGS-SLAM framework for RGB-D input. TCGSplat maintains a bounded, time-evolving confidence state for each Gaussian and updates it using rendering residuals and history-aware self-change feedback. The resulting confidence is rendered into image-space maps through Gaussian alpha compositing and used to guide tracking, mapping, and submap-level viewpoint selection. Experiments on Replica, TUM RGB-D, and ScanNet show that TCGSplat improves tracking robustness, maintains competitive

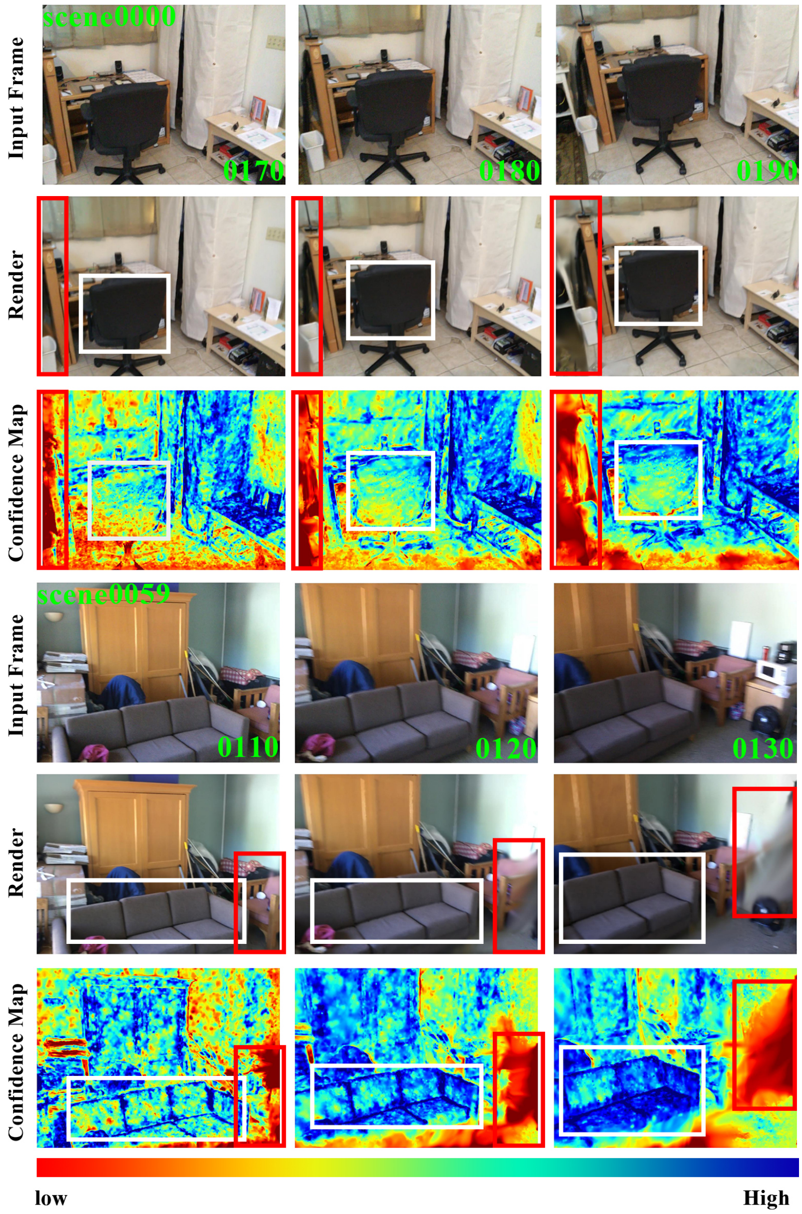

Fig. 3 Visualization of temporal confidence on ScanNet scenes 0000 and 0059. For each scene, columns correspond to different input frames, while rows show the input image, online rendering, and confidence map. White boxes indicate stable and well-reconstructed regions whose confidence increases over time, whereas red boxes mark newly observed regions with lower confidence.

reconstruction quality, and improves rendering quality while introducing moderate runtime overhead. These results demonstrate that temporal reliability modeling is a useful complement to geometry and appearance in online 3DGS-SLAM.

Declarations

- **Funding** This work was funded by Key Research and Development Program of Jilin Provincial Department of Science and Technology (20260201057GX).
- **Author contribution** Xingchen Guo conceived the preliminary idea, designed codes and experiments, and completed the first draft of the manuscript. Zhi Liu and Huilin Jiang provided theoretical guidance. Dandan Huang offered writing guidance and proofreading. Xingzhao Wang and Huiji Wang completed figure drawing.
- **Conflict of interest** The authors declare no conflict of interests.
- **Data availability** The TUM dataset is available at https://vision.in.tum.de/data/datasets/rgbd-dataset. The Replica dataset is available at https://github.com/facebookresearch/Replica-Dataset. The ScanNet dataset requires official application prior to download, and the application process is provided at https://github.com/ScanNet/ScanNet.
- **Code availability** The source code is available at https://github.com/gxcguxen/TCGSplat.

References

[1] Campos, C., Elvira, R., Rodríguez, J.J.G., M. Montiel, J.M., D. Tardós, J.: Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam. IEEE Transactions on Robotics **37**(6), 1874–1890 (2021) https://doi.org/10.1109/TRO.2021.3075644

[2] Newcombe, R.A., Izadi, S., Hilliges, O., Molyneaux, D., Kim, D., Davison, A.J., Kohi, P., Shotton, J., Hodges, S., Fitzgibbon, A.: Kinectfusion: Real-time dense surface mapping and tracking. In: 2011 10th IEEE International Symposium on Mixed and Augmented Reality, pp. 127–136 (2011). https://doi.org/10.1109/ISMAR.2011.6092378

[3] Whelan, T., Salas-Moreno, R.F., Glocker, B., Davison, A.J., Leutenegger, S.: Elasticfusion: Real-time dense slam and light source estimation. The International Journal of Robotics Research **35**(14), 1697–1716 (2016)

[4] Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: representing scenes as neural radiance fields for view synthesis. Commun. ACM **65**(1), 99–106 (2021) https://doi.org/10.1145/3503250

[5] Zhu, Z., Peng, S., Larsson, V., Xu, W., Bao, H., Cui, Z., Oswald, M.R., Pollefeys, M.: Nice-slam: Neural implicit scalable encoding for slam. In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 12776–12786 (2022). https://doi.org/10.1109/CVPR52688.2022.01245

[6] Johari, M.M., Carta, C., Fleuret, F.: Eslam: Efficient dense slam system based on hybrid representation of signed distance fields. In: 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 17408–17419 (2023). https://doi.org/10.1109/CVPR52729.2023.01670

[7] Sandström, E., Li, Y., Van Gool, L., Oswald, M.R.: Point-slam: Dense neural point cloud-based slam. In: 2023 IEEE/CVF International Conference on Computer Vision (ICCV), pp. 18387–18398 (2023). https://doi.org/10.1109/ICCV51070.2023.01690

[8] Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G., *et al.*: 3d gaussian splatting for real-time radiance field rendering. ACM Trans. Graph. **42**(4), 139–1 (2023)

[9] Yan, C., Qu, D., Xu, D., Zhao, B., Wang, Z., Wang, D., Li, X.: Gs-slam: Dense visual slam with 3d gaussian splatting. In: 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 19595–19604 (2024). https://doi.org/10.1109/CVPR52733.2024.01853

[10] Keetha, N., Karhade, J., Jatavallabhula, K.M., Yang, G., Scherer, S., Ramanan, D., Luiten, J.: Splatam: Splat, track & map 3d gaussians for dense rgb-d slam. In: 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 21357–21366 (2024). https://doi.org/10.1109/CVPR52733.2024.02018

[11] Yugay, V., Li, Y., Gevers, T., Oswald, M.R.: Gaussian-slam: Photo-realistic dense slam with gaussian splatting. arXiv preprint arXiv:2312.10070 (2023)

[12] Ji, Y., Liu, Y., Xie, G., Ma, B., Xie, Z., Liu, H.: Neds-slam: A neural explicit dense semantic slam framework using 3d gaussian splatting. IEEE Robotics and Automation Letters **9**(10), 8778–8785 (2024) https://doi.org/10.1109/LRA.2024.3451390

[13] Sandström, E., Zhang, G., Tateno, K., Oechsle, M., Niemeyer, M., Zhang, Y., Patel, M., Van Gool, L., Oswald, M.R., Tombari, F.: Splat-slam: Globally optimized rgb-only slam with 3d gaussians. In: 2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 1671–1682 (2025). https://doi.org/10.1109/CVPRW67362.2025.00156

[14] Zhu, L., Li, Y., Sandström, E., Huang, S., Schindler, K., Armeni, I.: Loopsplat: Loop closure by registering 3d gaussian splats. In: 2025 International Conference on 3D Vision (3DV), pp. 156–167 (2025). https://doi.org/10.1109/3DV66043.2025.00020

[15] Liso, L., Sandström, E., Yugay, V., Van Gool, L., Oswald, M.R.: Loopy-slam: Dense neural slam with loop closures. In: 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 20363–20373 (2024). https://doi.org/10.1109/CVPR52733.2024.01925

[16] Xu, Z., Li, Q., Liu, X., Chen, C., Niu, J.: Glc-slam: Robust loop closure for monocular gaussian splatting slam. Expert Systems with Applications **332**, 133585 (2027) https://doi.org/10.1016/j.eswa.2026.133585

[17] Hu, J., Chen, X., Feng, B., Li, G., Yang, L., Bao, H., Zhang, G., Cui, Z.: Cg-slam: Efficient dense rgb-d slam in a consistent uncertainty-aware 3d gaussian field. In: European Conference on Computer Vision, pp. 93–112 (2024). Springer

[18] Tran, A.T., Kosecka, J.: VarSplat: Uncertainty-aware 3D Gaussian Splatting for Robust RGB-D SLAM (2026). https://arxiv.org/abs/2603.09673

[19] Dufera, A.T., Cai, Y.-L.: Confidentsplat: Confidence-weighted depth fusion for accurate 3d gaussian splatting slam. In: 2025 IEEE 26th China Conference on System Simulation Technology and Its Applications (CCSSTA), pp. 911–917 (2025). https://doi.org/10.1109/IEEECONF65522.2025.11137090

[20] Guclu, O., Can, A.B.: Integrating global and local image features for enhanced loop closure detection in rgb-d slam systems: O. guclu, ab can. The Visual Computer **36**(6), 1271–1290 (2020)

[21] Sucar, E., Liu, S., Ortiz, J., Davison, A.J.: imap: Implicit mapping and positioning in real-time. In: 2021 IEEE/CVF International Conference on Computer Vision (ICCV), pp. 6209–6218 (2021). https://doi.org/10.1109/ICCV48922.2021.00617

[22] Cao, Z., Wang, Z., Zhang, Y., Jin, C., Cai, W., Zeng, Z., Liu, J.: Enhancing 3d gaussian splatting for low-quality images: semantically guided training and unsupervised quality assessment. The Visual Computer **41**(8), 5815–5833 (2025)

[23] Shen, J., Ruiz, A., Agudo, A., Moreno-Noguer, F.: Stochastic neural radiance fields: Quantifying uncertainty in implicit 3d representations. In: 2021 International Conference on 3D Vision (3DV), pp. 972–981 (2021). https://doi.org/10.1109/3DV53792.2021.00105

[24] Goli, L., Reading, C., Sellán, S., Jacobson, A., Tagliasacchi, A.: Bayes' rays: Uncertainty quantification for neural radiance fields. In: 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 20061–20070 (2024). https://doi.org/10.1109/CVPR52733.2024.01896

[25] Jiang, W., Lei, B., Daniilidis, K.: FisherRF: Active View Selection and Uncertainty Quantification for Radiance Fields using Fisher Information (2024). https://arxiv.org/abs/2311.17874

[26] Wang, H.: Submodular-based view selection for low-quality points rendering with multi-feature point-based nerf. The Visual Computer **41**(9), 6425–6437 (2025)

[27] Sandström, E., Ta, K., Van Gool, L., Oswald, M.R.: Uncle-slam: Uncertainty learning for dense neural slam. In: 2023 IEEE/CVF International Conference on

Computer Vision Workshops (ICCVW), pp. 4539–4550 (2023). https://doi.org/10.1109/ICCVW60793.2023.00488

[28] Wang, S., Xie, Y., Chang, C.-P., Millerdurai, C., Pagani, A., Stricker, D.: Uni-slam: Uncertainty-aware neural implicit slam for real-time dense indoor scene reconstruction. In: 2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), pp. 2228–2239 (2025). https://doi.org/10.1109/WACV61041.2025.00223

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

[30] Sturm, J., Engelhard, N., Endres, F., Burgard, W., Cremers, D.: A benchmark for the evaluation of rgb-d slam systems. In: 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 573–580 (2012). https://doi.org/10.1109/IROS.2012.6385773

[31] Dai, A., Chang, A.X., Savva, M., Halber, M., Funkhouser, T., Nießner, M.: Scannet: Richly-annotated 3d reconstructions of indoor scenes. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2432–2443 (2017). https://doi.org/10.1109/CVPR.2017.261

[32] Zhang, Y., Tosi, F., Mattoccia, S., Poggi, M.: Go-slam: Global optimization for consistent 3d instant reconstruction. In: 2023 IEEE/CVF International Conference on Computer Vision (ICCV), pp. 3704–3714 (2023). https://doi.org/10.1109/ICCV51070.2023.00345