FROSSv2: Improving Real-Time 3D Semantic Scene Graph Generation Model

Hao-Yu Hou, and Tzu-Yun Huang, and Yu-Shin Jian

Abstract—3D Semantic Scene Graphs provide structured representations essential for high-level scene understanding, yet existing methods often struggle to balance performance with real-time processing latency. While the recent FROSS framework achieved high speeds by simplifying the process of predicting 3D Gaussians, it suffered from geometric inaccuracies and merging conflicts due to simplified bounding-box lifting and strict class matching. Our work presents FROSSv2, an enhanced online 3D Semantic Scene Graph generation framework that addresses these limitations. We introduce two key contributions: incorporating the Segment Anything Model 2 to refine 2D-to-3D lifting via precise segmentation masks, and a novel hybrid merging criterion that leverages both spatial Gaussian Hellinger distance and semantic dot-product distance between class distributions. Extensive experiments on the 3DSSG and ReplicaSSG benchmark demonstrate that FROSSv2 significantly outperforms the baseline quantitatively and qualitatively, increasing relationship recall by 11.7%, object recall by 8.8% and predicate recall by 13.5%. Despite a marginal latency increase from 7 ms to 15 ms per frame, the system maintains robust real-time capabilities significantly faster than prior point-cloud-based and image-based approaches. Our implementation is publicly available at https://github.com/Howardkhh/FROSS/tree/v2.

Index Terms—3D Scene Graph, Gaussian, Real-Time

1 Introduction

Scene Graphs (SGs) [2] provide a structured and semantic representation of visual environments by modeling objects as nodes and their pairwise relationships as edges, as illustrated in Fig. 1. Such representations have proven effective for high-level vision tasks including image retrieval [2], generation [3], and reasoning [4], [5]. Extending this concept into 3D space, 3D Semantic Scene Graphs (SSGs) [6], [7] have emerged as a powerful framework for capturing both the geometric objects and semantic relationships within a scene. By jointly modeling object identities, spatial layouts, and interactions, 3D SSGs enable a wide range of applications in robotics and AR/VR systems, where spatial reasoning and decision-making are essential.

In real-world applications, however, 3D SSG generation systems must operate incrementally and under strict latency constraints. FROSS [1] addressed this need by proposing

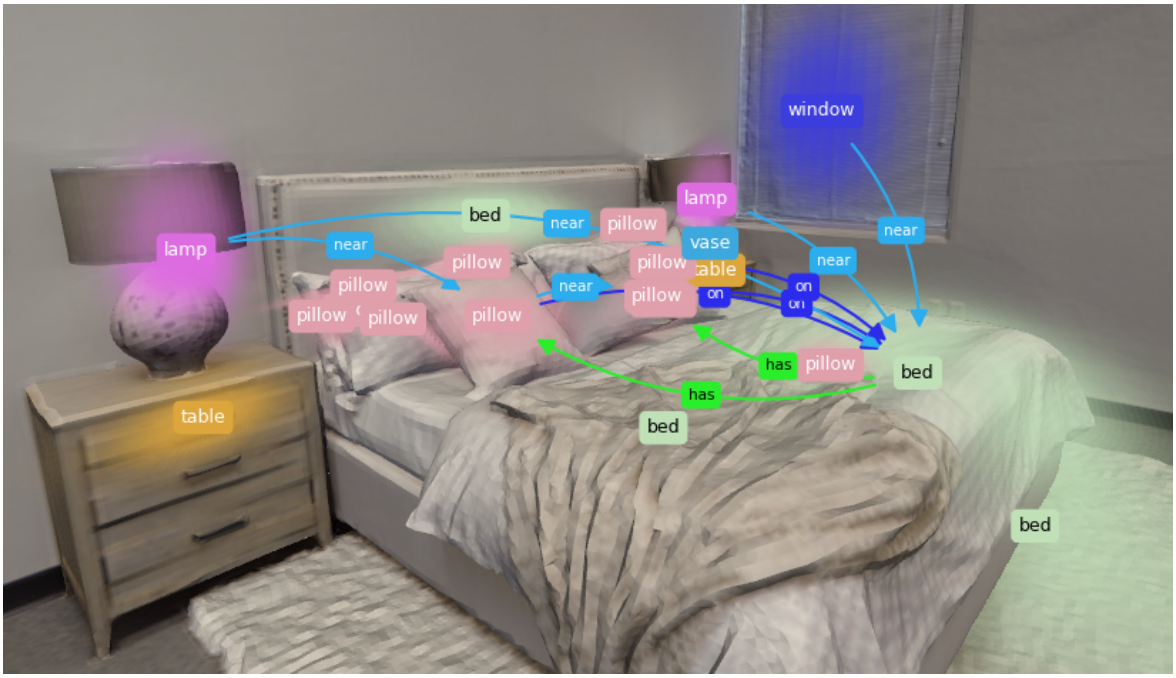

Fig. 1. An illustration of a 3D semantic scene graph, consisting of 3D object nodes and inter-object relationships. In the formulation of FROSS [1], the 3D objects are modeled as 3D Gaussian distributions.

a real-time and incremental 3D SSG generation pipeline that approximates objects using 3D Gaussian distributions, achieving faster-than-real-time performance at over 100 FPS. Despite its efficiency, our observations reveal several limitations that hinder its robustness. First, the original lifting procedure relies solely on 2D bounding box centers and corresponding depth values, which often leads to inaccurate 3D localization. Second, the variance along the depth axis of the 3D Gaussians is approximated using a naive heuristic, causing incorrect geometric estimation of thin objects. Lastly, FROSS's object merging mechanism strictly requires exact class labels matching and spatial similarity, frequently causing duplicated or fragmented nodes when predictions fluctuate across frames. These issues collectively degrade recall and reduce the reliability of long-term scene understanding.

To overcome these challenges, we enhance the FROSS pipeline with two key improvements. We first improve the 2D-to-3D lifting stage by integrating the Segment Anything Model 2 (SAM 2) [8] to obtain precise segmentation masks for each detected object. Combined with mask-based refinement and post-processing, this allows more accurate geometric estimation by utilizing all valid pixels rather than relying on noisy bounding box centers. Additionally, we propose a hybrid object merging criterion that jointly considers spatial Hellinger distance and semantic class distribution similarity. This design relaxes the overly strict class-matching rule in FROSS, enabling more stable and accurate merging across frames, especially when class predictions are uncertain.

Our contributions are summarized as follows:

1) A SAM 2 based 3D Gaussian lifting module that significantly improves geometric accuracy by refining 2D object regions and leveraging dense mask back-projection.
2) An improved hybrid merging algorithm that integrates spatial and semantic distribution similarity, reducing duplicate detections and increasing object and relationship recall.
3) Comprehensive quantitative and qualitative evaluations demonstrating FROSSv2's improvement over the original FROSS.

2 Related Work

Originally introduced by Johnson et al. [2], SGs provide a structured and semantic representation of visual scenes. Although initially proposed to improve the task of image retrieval [2], the ability of SGs to abstract complex visual information into condensed object-relation graphs has driven their adoption across a variety of computer vision tasks. These applications include image captioning [9], [10], image generation [3], and high-level scene understanding and reasoning [4], [5].

Building on the efficacy of 2D SGs, these graph-based representations was extended into the 3D domain to capture spatial geometry and semantic. Kim et al. [6] pioneered this extension with an incremental framework that constructs local graphs from RGB-D image streams, subsequently merging and updating them into a unified global graph. They also demonstrated the semantic utility of 3D SSGs by deploying them in 3D Visual Question Answering and robotic task planning scenarios. Following this foundational work, Wald et al. [7] introduced "3DSSG," the first large-scale dataset for 3D SSG evaluation. Derived from the 3RScan dataset [11], 3DSSG augments 3D scans with annotations for object attributes and inter-object relationships. Complementing the dataset, Wald et al. also presented a learned 3D SSG prediction method that operates on class-agnostic instance segmented point clouds. Further advancing incremental generation, Wu et al. [12], [13] proposed two successive methods on generating 3D SSGs from RGB-(D) sequences. Their approach first segments the incrementally reconstructed point clouds and subsequently predicts 3D object classes and relationships with a graph neural network. Together, these works have laid the foundation for research in the field of 3D SSGs.

Despite these advancements, processing latency remains a critical bottleneck. Most existing 3D SSG methods rely on point cloud processing, which is computationally intensive and often results in sub-real-time performance. To address this, FROSS [1] was proposed as a faster-than-real-time and online 3D SSG generation method. FROSS achieves speeds exceeding 100 frames-per-second (FPS) while maintaining competitive recall performance. This efficiency is primarily achieved by replacing dense point clouds with 3D Gaussian distributions to represent objects, thereby enabling instantaneous graph merging and updates between consecutive frames. The details of FROSS's method are presented in the following section.

3 Preliminary

3.1 Problem Definition

The primary problem concerned in FROSS [1] and this study is the online generation of 3D SSGs for environments where the complete scene structure remains unknown a priori. Given a sequence of input images, the primary objective is to construct a 3D SSG $\mathcal{G}$ of the target environment. The graph $\mathcal{G}$ consists of a set of nodes $\mathcal{V}$ and their corresponding directed edges $\mathcal{E}$. Each node $v_i = (c_i, \mu_i^{\text{3D}}, \Sigma_i^{\text{3D}}) \in \mathcal{V}$ is indexed by $i \in \{1, \ldots, N\}$, representing a unique object in the scene. Every object has a class label $c_i \in \mathcal{C}$, where $\mathcal{C}$ is the set of object categories. While different approaches may incorporate various node attributes, our formulation represents each object as a 3D Gaussian distribution parameterized by mean vector $\mu_i^{\text{3D}} \in \mathbb{R}^3$ and covariance matrix $\Sigma_i^{\text{3D}} \in \mathbb{R}^{3\times3}$. The edges $e_{i\to j} \in \mathcal{E}$ connect node v_i to node v_j with relationship $r_{i\to j} \in \mathcal{R}$, where $\mathcal{R}$ represents the set of predefined relationship types. Furthermore, the algorithm must be able to maintain online processing capabilities to update $\mathcal{G}$ continuously whenever new image data becomes available.

3.2 FROSS Framework

The top of Fig. 2 illustrates the overview of the FROSS pipeline. The framework takes RGB-D image sequences and camera poses as input to produce a coherent global 3D SSG. The pipeline operates sequentially through four stages:

(a) 2D SG Construction. For every incoming frame, FROSS first generates a local 2D SG. RT-DETR [14], [15] is

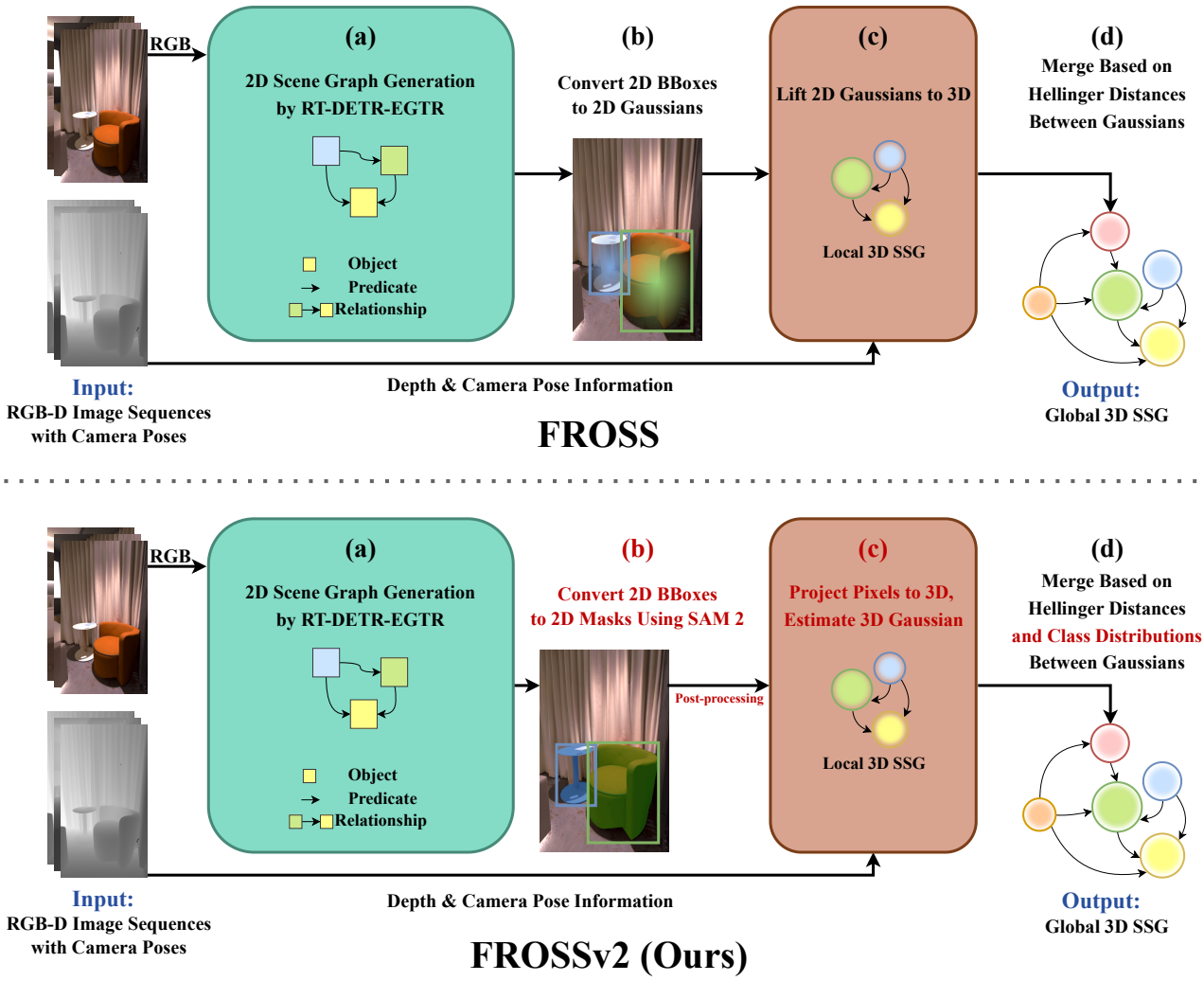

Fig. 2. An overview of the framework of original FROSS [1] (top) and FROSSv2 (bottom), with improvements marked in red. Panels (a)–(d) illustrate the main pipeline stages: 2D SG generation, 3D lifting, Gaussian estimation, and object merging, with FROSSv2 replacing (b)(c) via SAM 2 masks and improving (d) with a hybrid merging criterion.

utilized for real-time object detection, while the EGTR [16] relationship extractor is employed to predict semantic predicates between objects.

(b) 2D Gaussian Modeling. To facilitate the lifting of objects into 3D space, FROSS first models the detected 2D bounding boxes as Gaussian distributions. Specifically, FROSS treat each bounding box as a 2D uniform distribution, where the mean μ_i^{2D} corresponds to the box center. The covariance matrix Σ_i^{2D} is derived from the variance of a uniform distribution spanning the bounding box dimensions:

$$\Sigma_i^{2D} = \frac{1}{12}\begin{bmatrix} W_i^2 & 0 \\ 0 & H_i^2 \end{bmatrix}, \tag{1}$$

where W_i and H_i denote the width and height of the 2D bounding box, respectively.

(c) Lift 2D Gaussians to 3D. The 2D Gaussians are then lifted to 3D with the help of depth information and camera transformation matrices. First the 2D mean $\mu_i^{2D} := (p_x^{2D}, p_y^{2D})^\top$ is back-project to 3D through:

$$\mu_i^{3D} = RK^{-1}(p_x^{2D}, p_y^{2D}, 1)^\top \cdot z + \mathbf{t}, \tag{2}$$

where z corresponds to the depth value at μ_i^{2D}, $K \in \mathbb{R}^{3\times3}$ represents the camera intrinsic matrix, and $R \in \mathbb{R}^{3\times3}$ and $\mathbf{t} \in \mathbb{R}^3$ denote the rotation matrix and translation vector of the current camera, respectively. On the other hand, the 3D covariance is approximated using the Jacobian J of the local affine approximation of the projective transformation [17]. In the original literature of [17], the Jacobian J is utilized to project 3D covariance matrices onto the 2D image plane. FROSS's approach inverts this process by back-projecting the 2D covariance matrix Σ_i^{2D} into 3D space as Σ_i^{3D} via the pseudo-inverse J^+ of the Jacobian. However, simply inverting the equation results a 3D Gaussian that lacks variance information along the depth axis. FROSS hypothesizes that this variance approximates the average variance of the other dimensions, leading to the following equations:

$$\Sigma_i^{3D\prime\prime} = J^+\Sigma_i^{2D}J^{+\top}, \tag{3}$$

$$\Sigma_i^{3D\prime} = \Sigma_i^{3D\prime\prime} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \frac{(\Sigma_i^{3D\prime\prime})_{1,1}+(\Sigma_i^{3D\prime\prime})_{2,2}}{2} \end{bmatrix}, \tag{4}$$

$$\Sigma_i^{3D} = R\Sigma_i^{3D\prime}R^\top. \tag{5}$$

In this formulation, $(\Sigma_i^{3D\prime\prime})_{m,n} \in \mathbb{R}$ represents the element at position (m, n) of matrix $\Sigma_i^{3D\prime\prime}$.

(d) Merge and Update Global SSGs. With a local 3D SSG constructed from the current frame, it is integrated with the global 3D SSG through Gaussian merging and relationship update. A queue is initialized with nodes from the local graph. For each node v_i in this queue, FROSS calculates the Hellinger distance $H_D(i, j)$ between the Gaussian distributions of v_i and all other nodes v_j with the same semantic class ($c_i = c_j$). FROSS merges nodes with distance $H_D(i, j)$ below a threshold and adds unmerged nodes to the global 3D SSG as new objects. All relationship edges originally connected to v_i and v_j are then redirected to the newly merged node.

4 METHODOLOGY

4.1 Overview of FROSSv2 Framework

The bottom of Fig. 2 highlights the improvements introduced in FROSSv2. Building upon FROSS [1], FROSSv2 aims to improve both the accuracy of Gaussian approximation and the merging process. The main improvements of FROSSv2 are summarized as follows.

(b, c) 3D Gaussian Estimation with SAM 2. In FROSS, the 3D mean of each object is estimated using only the bounding box's center pixel and its corresponding depth. This approach often results in inaccurate back-projection, particularly when the object's center is occluded by another object. To address this issue, FROSSv2 uses SAM 2 [8] to generate segmentation masks for objects based on their bounding boxes, helping exclude irrelevant pixels within each bounding box for the projection step that follows. This enables a more accurate estimation of the object's true 3D position. Our improvement is detailed in Sec. 4.2.

(d) Hybrid Object Merging Criterion. The original FROSS merges objects only when they belong to the same class and exhibit high spatial similarity. This approach may overlook objects with similar class distributions when their most probable class labels do not match. Therefore, FROSSv2 adopts a hybrid merging algorithm that incorporates both spatial and class distribution similarity when determining whether two object nodes should be merged into the global 3D SSG. This new merging criterion is presented in Sec. 4.3.

4.2 3D Gaussian Estimation with SAM 2

FROSSv2 generates local 3D SSGs from 2D SGs by lifting 2D object bounding boxes into 3D Gaussian representations. This process involves two key steps:

1) Generating object masks with SAM 2 and refining them through two post-processing steps.
2) Back-projecting all mask pixels into 3D space using depth maps and camera transformations, then estimating a Gaussian distribution for each object.

Object Mask Generation with SAM 2 and Post-processing. We incorporate SAM 2 into our workflow to generate segmentation masks from 2D bounding boxes. Since the projected 3D Gaussians are estimated from the pixels in the masks along with their depth values and the corresponding camera transformations, it is essential to ensure that the pixels in a mask truly belong to the object. Therefore, we perform the following post-processing steps. First, for each pixel in a mask, we check whether it overlaps with masks of different object classes. If so, such ambiguous pixels are removed from every masks. This conservative filtering can avoid erroneous segmentation, and missing pixels can be recovered from other viewpoints. Next, each mask is eroded using a 7×7 kernel to further exclude potential background pixels. Finally, masks with fewer than 10 pixels are discarded. Here, FOREGROUNDPIXELCOUNT(m_i) denotes the number of valid (non-zero) pixels in mask m_i. The complete procedure is summarized in Algorithm 1.

Algorithm 1 Object Mask Generation with SAM 2 and Post-processing

Require: RGB image I, detected objects $O = \{(b_i, c_i)\}$
Ensure: Refined masks $M = \{m_i\}$

$M_{\text{raw}} \leftarrow \emptyset$
for all (b_i, c_i) in O **do**
  $m_i \leftarrow$ SAM2_GENERATEMASK(I, b_i)
  $M_{\text{raw}} \leftarrow M_{\text{raw}} \cup \{(m_i, c_i)\}$
end for
for all (m_i, c_i) in M_{raw} **do**
  for all pixel p in m_i **do**
    if p is inside some m_j with $c_j \neq c_i$ **then**
      mark p as ambiguous
    end if
  end for
  remove all ambiguous pixels from m_i
end for
for all (m_i, c_i) in M_{raw} **do**
  $m_i \leftarrow$ ERODE$(m_i, 7 \times 7)$
end for
$M \leftarrow \emptyset$
for all (m_i, c_i) in M_{raw} **do**
  if $|m_i| < 10$ **then**
    continue ▷ discard this mask
  end if
  $M \leftarrow M \cup \{(m_i, c_i)\}$
end for
return M

Mask Back-projection and 3D Gaussian Estimation. All valid pixels (p_x^{2D}, p_y^{2D}) in a mask are back-projected into 3D using the same equation as Eq. 2:

$$\mathbf{x}_i = RK^{-1}(p_x^{2D}, p_y^{2D}, 1)^\top \cdot z_{x,y} + \mathbf{t}, \tag{6}$$

where $z_{x,y}$ corresponds to the depth of the pixel (p_x^{2D}, p_y^{2D}). After obtaining the 3D points $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$, where $\mathbf{x}_i \in \mathbb{R}^3$, we estimate the parameters of the 3D Gaussian as follows. The mean vector is computed by

$$\mu^{3D} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{x}_i, \tag{7}$$

and the covariance matrix is given by

$$\Sigma^{3D} = \frac{1}{N} \sum_{i=1}^{N} (\mathbf{x}_i - \mu^{3D})(\mathbf{x}_i - \mu^{3D})^\top. \tag{8}$$

Here, μ^{3D} denotes the mean vector of the 3D Gaussian, and Σ^{3D} denotes its covariance matrix.

4.3 Merging 3D SSGs

Similar to FROSS, FROSSv2 integrates local 3D SSGs into a global 3D SSG through Gaussian merging and relationship accumulation. To determine whether two objects should be merged, FROSSv2 adopts a more robust merging criterion that considers both spatial and class-distribution similarity. Unlike FROSS, where each object is assigned a single class label $c_i \in \mathcal{C}$, FROSSv2 represents the class information as a distribution $c_i \in \mathbb{R}_+^{|\mathcal{C}|}$, where $|\mathcal{C}|$ is the number of class categories and $\sum_{k=1}^{|\mathcal{C}|} c_{i,k} = 1$.

Hybrid Criterion for Incremental Object Merging. For each object node v_i in a local SSG, FROSSv2 computes both the Hellinger distance $H_D(i,j)$ between the Gaussian distributions of v_i and all other nodes v_j in the local and global SSGs, and the class-distribution dot-product distance $\text{Dot}_D(c_i, c_j)$ between their class distributions c_i and c_j. If the weighted sum of these two distances falls below a threshold τ, the two nodes are regarded as the same object observed from different viewpoints and are merged. The class-distribution dot-product distance is defined as

$$\text{Dot}_D(c_i, c_j) = 1 - c_i^\top c_j = 1 - \sum_{k=1}^{|\mathcal{C}|} c_{i,k}\, c_{j,k} \tag{9}$$

Accordingly, two object nodes v_i and v_j are merged if

$$(1-w) \cdot H_D(i,j) + w \cdot \text{Dot}_D(c_i, c_j) < \tau, \tag{10}$$

where $w \in [0,1]$ is a predefined weight.

5 Experimental Results

5.1 Experimental Setup

5.1.1 Datasets

3DSSG. 3DSSG [7] is an extended 3D SSG dataset based on 3RScan, containing 1,482 semantically annotated scans of 478 different scenes. Each scan is complete with RGB-D sequences, 3D meshes, and instance segmentation. The dataset enriches this original data by adding object attributes and semantic relationships, such as "standing on" or "same color", making it a primary benchmark for 3D SSG evaluation. To address severe class imbalance of the dataset, we follow established protocols [12], [13], [18] by mapping the original labels to 20 NYUv2 [19] object categories and seven relationship types. While this results in a substantial collection of over 21,000 objects, the dataset's lower image quality and frame rate limit the performance of SAM 2 and the overall FROSSv2 pipeline.

ReplicaSSG. ReplicaSSG [1] extends the Replica [20] dataset with object relationship annotations. It features high-quality reconstructions of indoor 3D scenes with the ability of rendering videos from arbitrary trajectories. ReplicaSSG adopts object and relationship classes from the Visual Genome [21] dataset. Although the dataset is limited to only 18 scenes, we utilize this high-fidelity benchmark to conduct our qualitative experiments, allowing us to visually assess and validate the model's performance.

5.1.2 Evaluation Metric

To maintain consistency with existing literature and ensure fair comparisons with baseline methods, the primary evaluation metrics include Recall@K . These metrics are essential for quantifying the model's ability to retrieve accurate scene graph components from the data. *Object Recall* measures the proportion of ground truth objects that are successfully detected and assigned the correct semantic class label. Similarly, *Predicate Recall* assesses the accuracy of edge labels between detected objects, isolating the performance of object classification. *Relationship Recall* is the most comprehensive metric, evaluating the correctness of entire triplets, requiring correct detection and classification of the subject, the object, and the semantic predicate connecting them.

5.1.3 Matching Criteria for Predictions

The assessment of 3D SSG generation performance applies a rigorous strategy for matching detections with ground truths. Following protocols from FROSS [1], the evaluation relies on back-projected 3D points to establish correspondence, instead of generating explicit point cloud outputs. Points are derived from the projected segmentation masks from SAM 2 (our method) or bounding box (FROSS) and matched to the nearest ground truth point within a 0.1-meter radius. A detected object is considered a valid match if it satisfies two conditions:

1) There are more than 50% of its back-projected points must find their nearest neighbors within a specific ground truth object.
2) To ensure distinctiveness, the overlap ratio with the second-largest ground truth object must remain below a 75% threshold.

This strict one-to-one correspondence prevents over-segmentation artifact from inflating scores. Once objects are matched, relationship triplets (subject, object, predicate) are evaluated. A relationship is deemed correctly predicted only if both the subject and object nodes are accurately matched to the ground truth and the connecting predicate label is correctly classified.

5.1.4 Baseline Methods

To establish a strong baseline for our experiments, we reproduced the results of FROSS. We run the official source code and pre-trained models from the FROSS [1] implementation without additional training or fine-tuning. Additionally, we included comparisons with prior state-of-the-art methods, including 3DSSG [7] , SGFN [12] , Kim's framework [6], and Wu's framework [13]. For these methods, the quantitative performance metrics were directly adopted from the results reported in the FROSS literature.

5.2 FROSSv2 Implementation Details

For the FROSSv2 framework, the pre-trained model from the original FROSS is adopted, using RT-DETR and EGTR as the backbone for object detection and relationship extraction tasks. All hyperparameters were determined through grid search evaluation on the validation split, with particular emphasis on relationship performance. The parameter selection strategy prioritized configurations where all three

recall metrics exceeded the baseline, while ensuring precision remained similar to baseline levels. These optimal parameters were subsequently applied to the test split for final evaluation. Our implementation applies SAM 2 for mask generation, followed by a post-processing stage that applies a boundary erosion of 7 pixels. For the hybrid merging operations, the criteria is ruled by Equation 10, where the semantic weight w is set to 0.3 and the unified threshold τ is established at 0.7 to balance semantic and spatial consistencies between frames. The remaining hyperparameters follows those of FROSS [1].

5.3 Quantitative Results

Table 1 presents a comprehensive performance and latency comparison between the proposed FROSSv2, the original FROSS baseline, and various representative methodologies. The results reveal that FROSSv2 achieves the highest performance among all baseline methods. Specifically, FROSSv2 attains an object recall of 71.2%, a predicate recall of 46.5%, and a relationship recall of 39.6%. Compared to the baseline FROSS [1], this represents a substantial improvement of 8.8% in object detection, 13.5% in predicate classification, and 11.7% in relationship recall.

These significant gains prove the advantages of our method. While the integration of SAM 2 and the proposed hybrid merging criteria introduces a marginal increase in latency (from 7 ms to 15 ms), FROSSv2 remains significantly faster than prior real-time methods such as SGFN (161 ms) and Wu et al. (191 ms). The superior performance of FROSSv2 over point cloud-based reasoning methods further confirms the effectiveness of lifting scene graphs from 2D images.

TABLE 1
Comparison of 3D SSG generation methods on the 3DSSG dataset. Columns include input modalities and original reported end-to-end latency (omitting environmental mapping). The best results are highlighted in **red** and the second place in blue. All latencies are in milliseconds.

Method	Recall (%)			Latency	Input Modality
	Rel.	Obj.	Pred.		
3DSSG [7]	12.9	37.4	22.0	-	Point Cloud
SGFN [12]	22.0	51.6	27.5	161	RGB-D
Wu [13]	23.3	53.8	28.4	191	RGB-D
Kim [6]	9.1	59.0	7.1	310	RGB-D
FROSS [1]	27.9	62.4	33.0	**7**	RGB-D
FROSSv2 (Ours)	**39.6**	**71.2**	**46.5**	15	RGB-D

5.4 Qualitative Results

Fig. 3 presents a qualitative comparison between FROSS and FROSSv2 on the ReplicaSSG dataset. These examples highlight our method's ability to resolve common failure cases observed in the baseline. As illustrated in (a), FROSS fails to localize objects under occlusion. Specifically, the bounding box center of the sofa is occluded by the foreground vase, causing the baseline to back-project the sofa's Gaussian using an incorrect depth approximation of the vase. In contrast, FROSSv2 utilizes segmentation masks to filter out the occluding vase and plants, averaging the depth only across valid object pixels to lift the sofa to its correct 3D position.

Furthermore, (b) and (c) demonstrate FROSSv2's improved shape approximation. Panel (b) shows that the baseline fails to model thin, planar objects (like windows), often rendering them as amorphous blobs due to rough depth approximation during 2D-to-3D lifting. FROSSv2, however, accurately captures the window's planar structure with tighter Gaussians. Finally, in (c), our method avoids the surface projection artifacts seen in FROSS, where multiple Gaussians are incorrectly placed on the object's surface. Instead, FROSSv2 provides a more accurate approximation of the object's geometry (e.g., the yellow region enveloping the round ottoman). These visualizations confirm that utilizing segmentation masks enables a more faithful and robust 3D representation.

Additional video demonstrations of FROSSv2 are provided in this Google Drive link. In these videos, the upper half displays the 2D SG prediction for the current frame, while the lower half visualizes the global 3D SSG after merging. As new frames are processed, the global graph is progressively updated, with the Gaussian representations adjusting to better reflect the objects' geometry.

TABLE 2
Runtime analysis of the key components of FROSSv2. **Obj. Det.** refers to the object detection model RT-DETR [14], [15], while **Rel. Ext.** corresponds to the EGTR relationship extractor [16]. Only components with significant latency impact are included.

Mean Latency (in millisecond)			
Obj. Det.	**Rel. Ext.**	**SAM 2**	**Post-processing**
2.41	4.57	5.69	0.17
Back-projection	**3D Gaussian**	**Merging**	
0.23	0.59	0.37	

5.5 Runtime Analysis

Regarding computational efficiency, while the introduction of the SAM 2 segmentation model increases the processing load compared to the lightweight bounding-box lifting in FROSS, FROSSv2 maintains real-time capabilities. The runtime of each major component in FROSSv2 are shown in Table 2. The system operates at approximately 68.23 FPS on an NVIDIA GeForce RTX 4090 GPU, which is sufficient for online robotic applications. The results collectively establish FROSSv2 as a robust solution that balances high-level semantic accuracy with the efficiency required for online processing.

5.6 Ablation Experiment Results

To systematically evaluate the contributions of the proposed components in FROSSv2, we conduct a comprehensive ablation study focusing on two primary axes of improvement: the enhancement of 3D Gaussian geometric approximation via the SAM 2, and the robustness of the incremental merging mechanism through introducing of object class probability distributions.

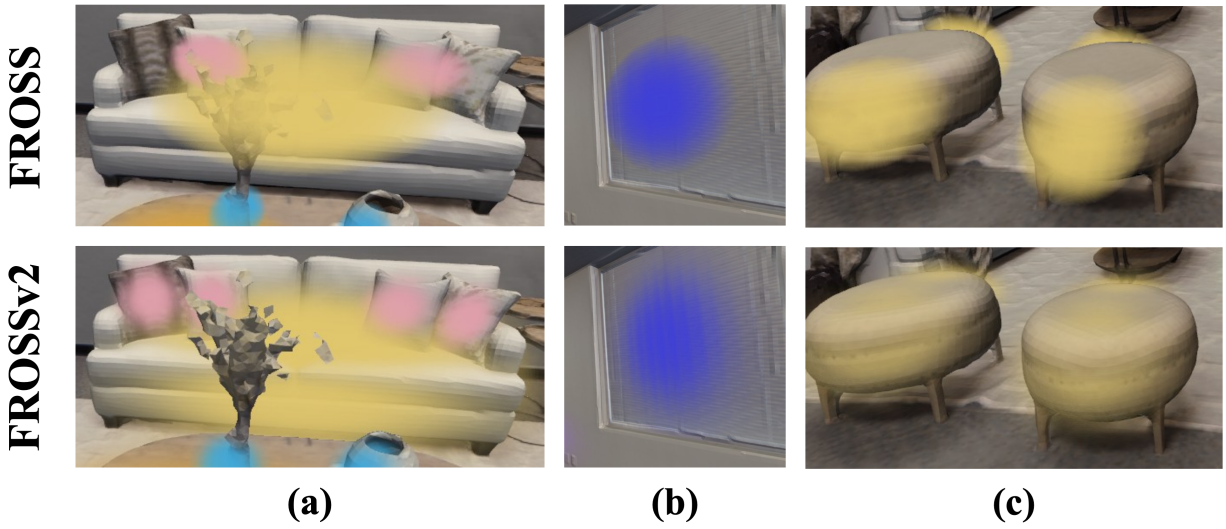

Fig. 3. Qualitative comparison between FROSS and FROSSv2 on ReplicaSSG dataset. The results demonstrating improvements of FROSSv2 over FROSS in challenging scenarios. (a) **Occlusion**: In FROSS, the sofa is incorrectly obscured by the vase and plant in the foreground, while FROSSv2 handles the depth better. (b) **Thin Structures**: Comparison of 3D Gaussian predictions on a window, representing a thin, planar object. (c) **Position Alignment**: FROSS exhibits artifacts where Gaussians drift off the object surface (see the yellow area on the round chair), while FROSSv2 shows tighter alignment to the geometry.

5.6.1 SAM 2 and Post-processing

The effectiveness of incorporating SAM 2 is first investigated to mitigate the limitations of the baseline FROSS framework, which approximates 3D objects by lifting 2D bounding boxes directly into 3D Gaussian distributions. While the baseline approach is computationally efficient, it frequently incorporates background noise into the object representation, particularly for non-rectangular instances. To address this, the integration of SAM 2 is evaluated for generating object masks prior to the lifting process. Although the direct application of SAM 2 improves boundary adherence compared to the bounding box baseline, raw prediction masks occasionally exhibit spatial overlaps between adjacent instances, which introduces ambiguity in the depth distribution of the lifted Gaussians. Consequently, a post-processing stage is introduced that removes overlapping mask regions and erodes boundaries. As presented in Table 3, this refinement significantly reduces noise in the 3D covariance estimation. The empirical results indicate that while SAM 2 alone provides a geometric improvement, the inclusion of post-processing is critical for generating precise local 3D SSGs by ensuring that the Gaussian statistics accurately reflect the core geometry of the objects.

5.6.2 Hybrid Merging Criterion

The merging mechanism is subsequently analyzed to address the challenge of consistent object merging across frames, particularly when the 2D object detector yields uncertain class predictions (e.g., a "sofa" misclassified as a "chair"). The original FROSS relied on a constraint of exact matching classes between two prediction, which prevents the merging of semantically similar but distinctly labeled instances.

To mitigate the effect of the original method, our merging strategies introducing predicted class probability distributions are investigated. Specifically, the dot-product distance of class distributions was prioritized as the semantic metric. Initially, a "dual thresholding" strategy was tested where a merge candidate must satisfy both a spatial Hellinger distance threshold and a semantic dot-product distance threshold independently. However, enforcing these two conditions simultaneously shows excessively stringent, leading to system instability and merging failures.

To resolve this, a hybrid merging criterion is proposed, which computes a weighted sum of the spatial Hellinger distance and the semantic dot product, governed by a unified single threshold. This formulation introduces a "soft" relaxation, allowing a strong spatial match to compensate for minor semantic discrepancies. Experimental results demonstrate that this hybrid approach significantly improves the robustness of object persistence and reduces object redundancy compared to the strict logic employed in the baseline.

TABLE 3
Ablation results for different components of FROSSv2. 'Post.' refers to the post-processing steps described in Sec. 4.2, and 'Hybrid Criterion' denotes the merging strategy introduced in Sec. 4.3.

Method	Recall (%)		
	Rel.	Obj.	Pred.
FROSS	27.9	62.4	33.0
FROSS + SAM 2 w/o Post.	32.6	64.3	39.2
FROSS + SAM 2 w/ Post.	34.8	69.2	40.9
FROSS + Hybrid Criterion	35.6	69.5	40.6
FROSSv2 (Ours)	**39.6**	**71.2**	**46.5**

6 Discussion

FROSSv2 achieves higher recall than prior work while maintaining an acceptable running time, demonstrating its

potential for practical use in systems requiring real-time 3D scene understanding. Nonetheless, several limitations and future works remain.

6.1 Limitations

Gaussian Representation Limitations for Irregularly Shaped Objects. Although 3D Gaussians effectively approximate an object's location and general extent, they struggle to represent objects with highly irregular geometry. For instance, a computer chair often contains thin or hollow structures. From different viewpoints, the chair may be segmented into small, fragmented masks, each producing a much smaller Gaussian. These Gaussians can be difficult to merge, resulting in multiple partial representations of the same object.

Low Precision Due to Over-Segmentation. Following the above issue, fragmented Gaussians may be interpreted as distinct objects in the merging stage. While this does not significantly affect recall, it reduces precision by producing scene graphs with more object nodes than actually exist in the environment.

Lack of a Deletion Mechanism. Once an object is detected in the 2D scene graph, it persists in the global scene graph indefinitely, regardless of whether later evidence contradicts the detection. As a result, false positives cannot be corrected. A possible solution would involve introducing a deletion mechanism that removes nodes that fail to reappear in subsequent frames. However, this strategy may also remove real objects that are small or hard to detect, which makes it difficult to design a deletion rule that works well in all cases.

6.2 Future Work

False Positive Reduction through Cross-Frame Consistency. Although FROSSv2 operates in static indoor environments where objects generally do not move, the current pipeline treats each frame independently when constructing local 2D SGs. Incorporating temporal consistency across consecutive frames may help identify and suppress false positives. For example, an object hypothesis that appears only once, or whose estimated size, shape, and position change drastically between adjacent frames, is likely unreliable. Leveraging multi-frame consistency checks could therefore improve robustness, particularly in cases where the detector produces occasional spurious predictions.

Learning-Based Merging Criteria. While the proposed hybrid merging criterion improves over the original FROSS, it still relies on manually defined distance metrics and hand-tuned hyperparameters. Moreover, the current merging process considers only spatial similarity and class-distribution similarity, omitting visual appearance cues. Designing a single rule that generalizes across all object categories is challenging, as objects vary widely in shape, size, and geometry. A promising future direction is to learn the merging function directly using a deep neural network. Such a model could take as input the mean vectors, covariance matrices, and visual features of the segmented regions, and output a probability indicating whether two nodes should be merged. This learning-based approach may better capture complex merging patterns and adapt to diverse object classes.

7 Conclusion

In this work, we presented FROSSv2, an enhanced real-time 3D SSG generation framework designed to address the geometric inaccuracies and merging inconsistencies observed in FROSS. By integrating SAM 2 and a hybrid merging criterion, FROSSv2 achieves substantially more accurate object approximation and a more reliable merging process through a soft coupling of spatial similarity and class-distribution similarity. Experiments on 3DSSG benchmark demonstrate that FROSSv2 outperforms prior methods, achieving notable gains of 8.8% in object recall, 13.5% in predicate recall, and 11.7% in relationship recall, while maintaining real-time processing at 15 ms per frame. Overall, FROSSv2 demonstrates that combining precise segmentation with distribution-aware object merging provides a promising path toward accurate, scalable, and real-time 3D scene understanding.

References

[1] H.-Y. Hou, C.-Y. Lee, M. Sonogashira, and Y. Kawanishi, "FROSS: Faster-than-Real-Time Online 3D Semantic Scene Graph Generation from RGB-D Images," in Int. Conf. Comput. Vis., October 2025.

[2] J. Johnson, R. Krishna, M. Stark, L.-J. Li, D. Shamma, M. Bernstein, and L. Fei-Fei, "Image Retrieval using Scene Graphs," in IEEE Conf. Comput. Vis. Pattern Recog., 2015, pp. 3668–3678.

[3] J. Johnson, A. Gupta, and L. Fei-Fei, "Image Generation from Scene Graphs," in IEEE Conf. Comput. Vis. Pattern Recog., 2018, pp. 1219–1228.

[4] L. Li, Z. Gan, Y. Cheng, and J. Liu, "Relation-Aware Graph Attention Network for Visual Question Answering," in Int. Conf. Comput. Vis., 2019, pp. 10 313–10 322.

[5] T. Qian, J. Chen, S. Chen, B. Wu, and Y.-G. Jiang, "Scene Graph Refinement Network for Visual Question Answering," IEEE Trans. Multimedia, vol. 25, pp. 3950–3961, 2022.

[6] U.-H. Kim, J.-M. Park, T.-J. Song, and J.-H. Kim, "3-D Scene Graph: A Sparse and Semantic Representation of Physical Environments for Intelligent Agents," IEEE Trans. Cybernetics, vol. 50, no. 12, pp. 4921–4933, 2019.

[7] J. Wald, H. Dhamo, N. Navab, and F. Tombari, "Learning 3D Semantic Scene Graphs from 3D Indoor Reconstructions," in IEEE Conf. Comput. Vis. Pattern Recog., 2020, pp. 3961–3970.

[8] N. Ravi, V. Gabeur, Y.-T. Hu, R. Hu, C. Ryali, T. Ma, H. Khedr, R. Rädle, C. Rolland, L. Gustafson, E. Mintun, J. Pan, K. V. Alwala, N. Carion, C.-Y. Wu, R. Girshick, P. Dollár, and C. Feichtenhofer, "SAM 2: Segment Anything in Images and Videos," arXiv:2408.00714, 2024. [Online]. Available: https://arxiv.org/abs/2408.00714

[9] L. Gao, B. Wang, and W. Wang, "Image Captioning with Scene-Graph Based Semantic Concepts," in Int. Conf. on Mach. Learn. Comput., 2018, pp. 225–229.

[10] X. Li and S. Jiang, "Know More Say Less: Image Captioning based on Scene Graphs," IEEE Transactions on Multimedia, vol. 21, no. 8, pp. 2117–2130, 2019.

[11] J. Wald, A. Avetisyan, N. Navab, F. Tombari, and M. Niessner, "RIO: 3D Object Instance Re-Localization in Changing Indoor Environments," in Int. Conf. Comput. Vis., 2019.

[12] S.-C. Wu, J. Wald, K. Tateno, N. Navab, and F. Tombari, "SceneGraphFusion: Incremental 3D Scene Graph Prediction from RGB-D Sequences," in IEEE Conf. Comput. Vis. Pattern Recog., 2021, pp. 7515–7525.

[13] S.-C. Wu, K. Tateno, N. Navab, and F. Tombari, "Incremental 3D Semantic Scene Graph Prediction from RGB Sequences," in IEEE Conf. Comput. Vis. Pattern Recog., 2023, pp. 5064–5074.

[14] Y. Zhao, W. Lv, S. Xu, J. Wei, G. Wang, Q. Dang, Y. Liu, and J. Chen, "DETRs Beat YOLOs on Real-Time Object Detection," in IEEE Conf. Comput. Vis. Pattern Recog., 2024, pp. 16 965–16 974.
[15] W. Lv, Y. Zhao, Q. Chang, K. Huang, G. Wang, and Y. Liu, "RT-DETRv2: Improved Baseline with Bag-of-Freebies for Real-Time Detection Transformer," arXiv:2407.17140, 2024. [Online]. Available: https://arxiv.org/abs/2407.17140
[16] J. Im, J. Nam, N. Park, H. Lee, and S. Park, "EGTR: Extracting Graph from Transformer for Scene Graph Generation," in IEEE Conf. Comput. Vis. Pattern Recog., 2024, pp. 24 229–24 238.
[17] M. Zwicker, H. Pfister, J. Van Baar, and M. Gross, "EWA Splatting," IEEE Trans. Vis. Comput. Graph., vol. 8, no. 3, pp. 223–238, 2002.
[18] J. Wald, N. Navab, and F. Tombari, "Learning 3D Semantic Scene Graphs with Instance Embeddings," Int. J. Comput. Vis., vol. 130, no. 3, pp. 630–651, 2022.
[19] P. K. Nathan Silberman, Derek Hoiem and R. Fergus, "Indoor Segmentation and Support Inference from RGBD Images," in Eur. Conf. Comput. Vis., 2012.
[20] J. Straub, T. Whelan, L. Ma, Y. Chen, E. Wijmans, S. Green, J. J. Engel, R. Mur-Artal, C. Ren, S. Verma, A. Clarkson, M. Yan, B. Budge, Y. Yan, X. Pan, J. Yon, Y. Zou, K. Leon, N. Carter, J. Briales, T. Gillingham, E. Mueggler, L. Pesqueira, M. Savva, D. Batra, H. M. Strasdat, R. D. Nardi, M. Goesele, S. Lovegrove, and R. Newcombe, "The Replica Dataset: A Digital Replica of Indoor Spaces," arXiv:1906.05797, 2019.
[21] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma et al., "Visual Genome: Connecting Language and Vision using Crowdsourced Dense Image Annotations," Int. J. Comput. Vis., vol. 123, pp. 32–73, 2017.