REVISABLE ONLINE 3D SCENE GRAPH GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Online 3D semantic scene graph generation must assign frame-level detections to persistent object nodes while observations are partial and noisy. Once made, these assignments and the resulting relation endpoints are usually fixed, so early false merges and splits persist throughout processing. We introduce Revisable 3D Scene Graph Generation (ReSGG), an online setting that allows historical observation-to-node assignments to change as new views provide evidence. We propose *SGReviser*, a training-free back end for existing scene graph front ends. SGReviser detects local association conflicts and compares joint and separate surface reconstructions using leave-one-frame-out geometric evidence. It converts the resulting signed preferences into an observation partition with a constrained correlation-clustering resolver, abstaining when the evidence is inconclusive. Because relation records retain their observation-level endpoints, graph edges can be repaired after repartitioning without re-predicting predicates. On 3DSSG, SGReviser improves relation Recall@1 by 16.4 points for FROSS and 13.0 points for NoPA; on ReplicaSSG, it improves the corresponding scores by 4.6 and 5.7 points. The method operates online with modest latency and requires no task-specific training, demonstrating that revising historical associations can substantially improve online 3D scene graph quality.

1 INTRODUCTION

3D Scene Graphs (3DSG) represent environments with object nodes and semantic relation edges, providing embodied agents with structured representations for navigation and manipulation (Armeni et al., 2019; Kim et al., 2020; Rana et al., 2023; Gu et al., 2024; Yin et al., 2024). Unlike offline reconstruction, online 3D scene graph generation builds this representation incrementally from partial RGB-D streams (Wu et al., 2023; Bickici et al., 2026). Each frame provides incomplete observations, updates object geometry and semantics, and predicts relations under time and memory constraints. Because each view reveals only part of an object, assigning an observation to a node is uncertain. We call a frame-level detection an *observation*, the observations assigned to one object a *node*, and an edge between two nodes a *relation edge* labeled by a predicate.

These assignments must be made before disambiguating evidence arrives. From one viewpoint, neighboring chairs may be falsely fused (Fig. 1a), while occlusion or noise may split one table across nodes. Such errors propagate to relations: false merges mix observations and aggregate their predicate records into incorrect edges. Later views may reveal the mistake but cannot undo it. Correcting the graph therefore requires regrouping stored observations and remapping affected records to revised endpoints (Fig. 1c). This motivates our central question: *how can an online system produce a graph at every frame while keeping historical associations revisable?*

Representative online models include SceneGraphFusion (Wu et al., 2021) and MonoSSG (Wu et al., 2023), which update geometric and semantic states frame by frame, as well as FROSS (Hou et al., 2025) and NoPA (Yeo et al., 2026), which accelerate this process with compact object representations. PUF (Yang et al., 2026) models uncertainty during fusion. These methods assign each observation to a node when it arrives and use that assignment for subsequent fusion and relation prediction, but do not move historical observations between nodes (Fig. 1b). Other mapping methods, including Fusion++ (McCormac et al., 2018), Voxblox++ (Grinvald et al., 2019), and Panoptic Multi-TSDFs (Schmid et al., 2022), maintain object instances and merge or discard hypotheses as evidence accumulates. Hydra (Hughes et al., 2022) and Khronos (Schmid et al., 2024) revise maps after loop closure. These methods primarily revise volumetric states, object hypotheses, or poses;

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

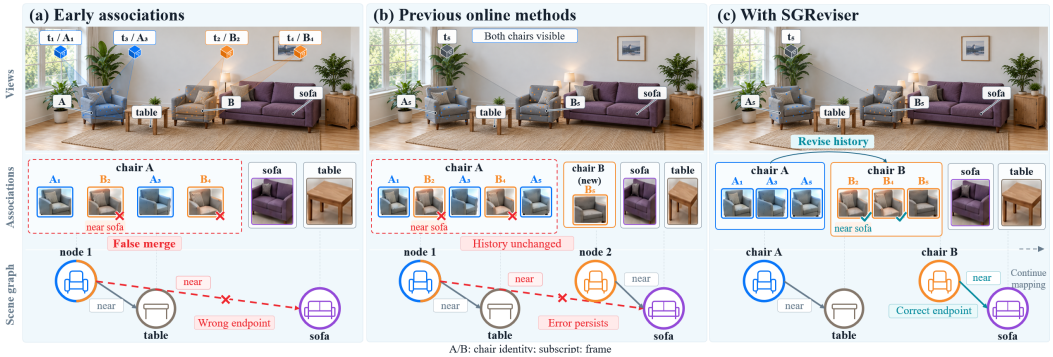

Figure 1: Revising historical associations. (a) Early views falsely merge two chairs. (b) Later views add a new node but preserve incorrect historical assignments and relation endpoints. (c) SGREVISER regroups the observations and repairs the affected relations.

relation records do not generally follow revised identities. The missing capability is a geometric test for revising historical associations in an online scene graph. Point-set overlap is insufficient, i.e., different views of one object may expose complementary surfaces with little overlap, while nearby objects may overlap substantially.

We formulate *Revisable 3D Scene Graph Generation* (RESGG), an online setting where previously assigned observations and relation records can be reassigned as later views reveal new evidence. We propose SGREVISER, a revision method for RESGG based on a held-out-view comparison. Given two candidate observation groups, it reconstructs their surfaces while excluding an evaluation frame, then compares two hypotheses: a joint surface pooling both groups and separate surfaces for each group. Pooling complementary views of one object can better explain the held-out observation, whereas pooling different objects may place geometry outside the target mask or observed depth. Excluding the evaluation frame prevents an observation from explaining itself. The difference in hypothesis scores provides geometric evidence under the supplied poses and depth, rather than a calibrated identity probability. When views are insufficient or the hypotheses are similarly supported, SGREVISER abstains instead of forcing a merge or split.

SGREVISER serves as a plug-and-play module for existing online 3D SGG methods. A lightweight trigger reopens a node when a new observation conflicts with its historical support or when two observations from the same frame are assigned to it. The revision scope includes the flagged node and alternative nodes ranked by the front end, allowing both sides of a possible false split to be considered. Observations in this scope are grouped into temporal fragments for efficient geometric testing, while reassignment remains at the observation level. Held-out comparisons yield signed preferences to merge or separate fragments and abstain when the evidence is inconclusive. A constrained correlation-clustering step (Bansal et al., 2004) converts these preferences into a feasible observation partition under same-frame exclusions. Because the resolver starts from individual observations, it can both split false merges and join false splits. The proposed partition is accepted only if it agrees better with the evidence than the current grouping or repairs a same-frame violation. Since each predicate record retains its endpoint observations, the revised partition directly determines which relation edges must be reconnected. Small revisions run synchronously, whereas larger ones run in the background. SGREVISER requires no additional task-specific training.

We attach SGREVISER to FROSS and NoPA and evaluate it on 3DSSG and ReplicaSSG. Relation Recall@1 improves by 16.4 and 13.0 percentage points over FROSS and NoPA on 3DSSG, and by 4.6 and 5.7 points on ReplicaSSG. Streaming relation recall improves in all four comparisons, with mean per-frame latency between 20.57 and 26.78 ms. These results support retrospective association as a complement to fast online scene graph models.

Contributions. We formulate RESGG as revisable online scene graph generation, where stored observations and relation records can be regrouped as new evidence arrives. We introduce a held-out-view comparison of joint and separate reconstructions that produces signed geometric evidence and abstains when evidence is insufficient. We embed this evidence in a constrained revision procedure that runs synchronously or in the background, and show on two front ends and two benchmarks that it improves final and streaming scene graph quality without additional task-specific training.

2 Problem Formulation and Geometric Evidence

ReSGG problem definition. Revisable 3D Scene Graph Generation (ReSGG) builds a scene graph from an RGB-D stream while allowing previously committed observation-to-node assignments to change when later views provide new evidence. We consider a static scene observed with known camera intrinsics, camera poses, and depth measurements. The main experiments use benchmark poses and rendered depth. At frame t, let $\mathcal{E}_t = \{e_i \mid t_i \le t\}$ denote all observations received so far. Each observation $e_i = (t_i, M_i, X_i, p_i)$ contains a frame index, a detection region, 3D points expressed in a common scene coordinate system by transforming valid depth points with the supplied camera poses, and a predicted category. We use detection boxes for M_i, so X_i may include background points.

Before revision at frame t, the front end provides a graph Γ_t^0 with node set $\mathcal{V}_t^0$ and assignment $a_t^0 : \mathcal{E}_t \to \mathcal{V}_t^0$, where the superscript 0 denotes the pre-revision state. Node k aggregates $C_k^0 = \{e_i \in \mathcal{E}_t \mid a_t^0(e_i) = k\}$ into its geometry and category. We also retain observation-level relation records $(e_i, r, e_j) \in \mathcal{R}_t$, where r is a predicted predicate between two observations. The records supporting the directed edge from node k to node h are those satisfying $a_t^0(e_i) = k$ and $a_t^0(e_j) = h$. Because each record preserves its original observation endpoints, a revision can update node assignments and relation endpoints without rerunning predicate prediction.

Revision as constrained repartitioning. Revision operates on a local set $\mathcal{S}_t \subseteq \mathcal{E}_t$ containing the complete histories of the participating nodes (Sec. 3.1). Let P_0 denote the partition of $\mathcal{S}_t$ induced by a_t^0, and let $s_{ij}(P) \in \{0, 1\}$ indicate whether observations e_i and e_j belong to the same group in partition P. The number of groups is unrestricted, so a revision may split or merge nodes. Observations remain indivisible, assignments outside $\mathcal{S}_t$ stay fixed, and relation edges incident to revised observations are reaggregated afterward.

We assume that distinct observations from the same frame represent different instances. The corresponding cannot-link pairs (Wagstaff et al., 2001) are $\mathcal{H} = \{(i, j) \mid e_i, e_j \in \mathcal{S}_t,\ i < j,\ t_i = t_j\}$, and a feasible partition $P \in \mathfrak{P}_{\mathcal{H}}$ must satisfy $s_{ij}(P) = 0$ for every $(i, j) \in \mathcal{H}$. If this assumption holds, an infeasible P_0 indicates a false merge within the revision scope. Duplicate detections violate the assumption and may therefore be forced apart even when they correspond to one instance. In contrast, an observation spanning multiple objects cannot be separated because observations are indivisible. These constraints rule out impossible assignments but do not determine which observations from different frames belong together. We therefore use signed geometric preferences from multiple views to select a feasible partition (Sec. 3).

Leave-one-frame-out surfaces and depth consistency. Direct point overlap can miss complementary views of one object and confuse nearby objects. We instead evaluate reconstructed surfaces against observations excluded from their construction. For an observation set A and evaluation frame f, define $S_A^{-f} = \mathrm{Voxel}\left(\bigcup_{\substack{e_i \in A \\ t_i \neq f}} X_i\right)$, where voxelization retains one centroid per occupied voxel of side length s. Thus, all observations from frame f are excluded from the surface being evaluated. For a singleton set $A = \{e_j\}$, this reduces to the voxelized points of e_j whenever $t_j \neq f$.

Rendering S into frame f with z-buffering and splat radius r produces depth $\hat{d}_f^S(u)$. We compare it with the input depth $D_f(u)$ on $\Omega_f(S)$, the pixels where both depths are valid and which lie at least b pixels from discontinuities in the input depth map. With tolerance δ, depth-consistent pixels are $V_f(S) = \{u \in \Omega_f(S) : |\hat{d}_f^S(u) - D_f(u)| \le \delta\}$, free-space contradictions are $F_f(S) = \{u \in \Omega_f(S) : \hat{d}_f^S(u) < D_f(u) - \delta\}$, and occluded pixels are $B_f(S) = \{u \in \Omega_f(S) : \hat{d}_f^S(u) > D_f(u) + \delta\}$. Geometry behind the measured surface may be occluded, whereas geometry in front of it contradicts the observed free space (Curless & Levoy, 1996). We therefore exclude $B_f(S)$ and evaluate only $O_f(S) = V_f(S) \cup F_f(S)$.

Geometric consistency score. A surface–observation pair is testable, denoted by $\chi(S, e_i) = 1$, when $|X_i| \ge n_{\mathrm{pt}}$ and $|O_{t_i}(S)| \ge \max(n_{\mathrm{px}}, \beta|M_i|)$. Untestable pairs are omitted, whereas testable zero scores remain evidence. For a testable pair, coverage is $c(S, e_i) = |\{x \in X_i \mid d(x, S) \le \delta\}|/|X_i|$, mask precision is $\nu(S, e_i) = |V_{t_i}(S) \cap M_i|/|V_{t_i}(S)|$, and the free-space contradiction rate is $\phi(S, e_i) = |F_{t_i}(S)|/|O_{t_i}(S)|$, where $d(x, S)$ is the nearest-point distance.

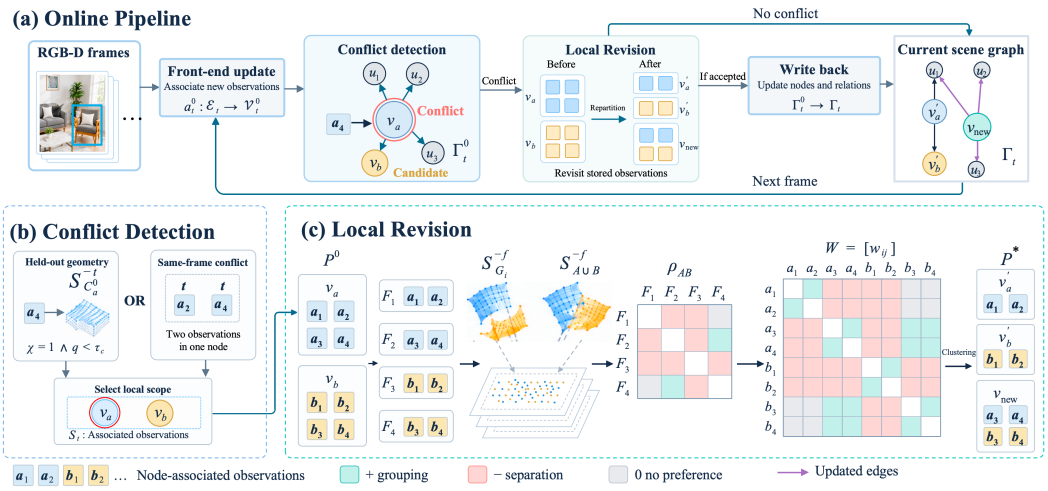

Figure 2: **Overview of SGREVISER.** A conflict reopens a local set of historical observations. Leave-one-frame-out reconstruction compares temporary fragments and produces signed association evidence. A constrained partition then reassigns individual observations, and stored relation provenance updates object nodes and relation endpoints.

We combine these quantities into $q(S, e_i) = (1 - \phi(S, e_i)) \frac{2c(S,e_i)\nu(S,e_i)}{c(S,e_i)+\nu(S,e_i)}$, setting $\nu(S, e_i)$ and the harmonic mean to zero when their denominators vanish. A large surface may cover all observed points but spill outside the detection region, yielding low precision; a small surface may remain inside the region but provide low coverage. The harmonic mean requires both terms to be high, while $1-\phi$ discounts free-space contradictions. For more parameter settings, please refer to App. E.

3 METHOD

SGREVISER, as shown in Figure 2, makes scene graphs revisable through four stages. It detects a local conflict and selects a revision scope; constructs temporary fragments and compares joint and separate leave-one-frame-out surfaces; converts the resulting signed evidence into an observation-level partition; and propagates the accepted assignment to nodes and relation edges.

3.1 CONFLICT TRIGGER AND LOCAL REVISION SCOPE

Most online assignments are correct, while the cost of revising a region grows with the number of observations it contains. Re-solving the full history after every frame would therefore spend most of its computation on assignments that need no change. We separate revision into a lightweight trigger and a more expensive resolver. The trigger identifies when a revision may be needed; the resolver determines which observations should be regrouped. The trigger mainly detects false merges, which leave geometric inconsistencies in the node that contains them.

Conflict trigger. After processing frame t, we inspect only nodes that received new observations. For a new observation e_i assigned to node k, we construct the node surface from C_k^0 while excluding frame t_i. We flag a geometric conflict when the pair is testable and $q(S_{C_k^0}^{-t_i}, e_i) < \tau_c$, that is, when $\chi(S_{C_k^0}^{-t_i}, e_i) = 1$ but the geometric consistency score falls below the trigger threshold. Excluding frame t_i prevents e_i from helping construct the surface against which it is evaluated. We also trigger revision when a node contains two observations from the same frame, which violates the same-frame cannot-link constraint defined in Sec. 2. A trigger only reopens the assignment but it does not determine the correction.

Local scope. Using only the flagged node would leave two common errors unresolved. In a false split, the two nodes may each appear internally consistent, so the second node would never be selected by the trigger. In a false merge, observations removed from the flagged node would have

no existing node to rejoin. We therefore add the top-K_{cand} active alternatives ranked by the front end for the conflicting observation. Let $\mathcal{V}_t^{\text{loc}}$ contain the flagged node and these alternatives. The revision scope is $\mathcal{S}_t = \bigcup_{h \in \mathcal{V}_t^{\text{loc}}} C_h^0$, with initial partition $P_0 = \{C_h^0\}_{h \in \mathcal{V}_t^{\text{loc}}}$. Only observations in $\mathcal{S}_t$ may change. The scope is small in the number of nodes, but each participating node contributes its complete history, so assignments made many frames earlier remain accessible.

3.2 Leave-One-Frame-Out Evidence for Association

The resolver needs signed evidence about whether two groups of observations depict the same object. Direct point overlap is unreliable: different views of one object may expose complementary surfaces with little overlap, while nearby objects may occupy similar regions in space. We instead ask whether pooling two groups improves their ability to explain an observation that was not used to build the pooled surface. Comparing every observation pair would be quadratic in history length, so we first construct temporal fragments and perform the geometric test between fragments.

Temporal fragments. A temporal fragment is a temporary group of observations formed by matching adjacent frames. It is used to construct geometric evidence and is not treated as an indivisible object node. Consecutive frames usually observe an object from nearby viewpoints, making adjacent-frame matching both efficient and reliable. For consecutive nonempty frames $f < f'$, we score each earlier observation e_i against each later observation e_j with $Q_{ij} = q(S^{-f'}_{\{e_i\}}, e_j)$ whenever the pair is testable. The Hungarian algorithm (Kuhn, 1955; Munkres, 1957) then finds the one-to-one partial matching $\mathcal{M}^*_{ff'}$ that maximizes $\sum_{(i,j)\in\mathcal{M}} Q_{ij}$ subject to $Q_{ij} \geq \tau_a$ for every matched pair. The threshold prevents a weak pair from being accepted merely to complete the matching. Matched pairs become undirected edges with weight $\alpha_{ij} = Q_{ij}$, and connected components of these edges within $\mathcal{S}_t$ form the fragments. We write G_i for the fragment containing observation e_i. Fragments provide surfaces and comparison units, but the final reassignment remains at the observation level, so a fragment can still be split by the resolver.

Joint and separate hypotheses. Consider two fragments A and B and an evaluation frame f. Under the joint hypothesis, observations are explained by the pooled surface $S^{-f}_{A\cup B}$. Under the separate hypothesis, an observation e_i is explained by the surface of its own fragment, $S^{-f}_{G_i}$. Both surfaces exclude every observation from frame f, so the comparison evaluates prediction on unseen evidence rather than fit to the construction data. Pooling complementary views of one object can improve coverage and mask precision. Pooling different objects can introduce geometry outside the target region or in observed free space and reduce the consistency score.

Rendering is the dominant cost, so we evaluate at most K_{check} frames per fragment, prioritizing conflict frames and then frames that add the most surface coverage. Let $\mathcal{I}_f^{AB}$ denote the observations from $A \cup B$ in frame f for which both hypotheses are testable, and let $\mathcal{F}_{AB}$ denote the evaluated frames with $\mathcal{I}_f^{AB} \neq \varnothing$. We define the pooling gain as $\Delta_{AB} = \frac{1}{|\mathcal{F}_{AB}|} \sum_{f\in\mathcal{F}_{AB}} \frac{1}{|\mathcal{I}_f^{AB}|} \sum_{e_i\in\mathcal{I}_f^{AB}} \left[q\big(S^{-f}_{A\cup B}, e_i\big) - q\big(S^{-f}_{G_i}, e_i\big) \right]$. We average observations within each frame before averaging across frames, so a single cluttered frame cannot determine the result solely through its larger number of observations.

Signed evidence and abstention. Leave-one-frame-out geometry can be inconclusive, i.e., occlusion may hide the surfaces needed to distinguish two objects, while redundant views of one object may provide little additional benefit from pooling. We convert the continuous pooling gain Δ_{AB} into a signed preference only when it has sufficient support and a clear margin: $\rho_{AB} = \text{sign}(\Delta_{AB})\mathbf{1}\left[|\mathcal{F}_{AB}| \geq m \ \wedge \ |\Delta_{AB}| \geq \eta\right]$, where $m \geq 2$. A value of $+1$ favors pooling, -1 favors separation, and 0 denotes insufficient evidence. The zero case is abstention rather than a separation vote: the comparison is omitted from the clustering objective, while same-frame exclusions remain enforced. Two singleton fragments share no held-out frame and therefore always abstain.

3.3 From Signed Evidence to a Constrained Partition

Fragment-level preferences do not directly define the final observation partition. They may be inconsistent, for example when A should join B, B should join C, but A and C should remain separate. The partition must also satisfy same-frame cannot-link constraints, while the number of objects in

the revision scope is unknown. We therefore use correlation clustering (Bansal et al., 2004), which partitions signed evidence without fixing the number of groups and supports hard constraints (van Zuylen & Williamson, 2009).

Lifting fragment evidence. We distribute fragment-level preferences over observation pairs as

$$w_{ij} = \mathbf{1}[G_i = G_j]\frac{\alpha_{ij}}{Z_{G_i}} + \mathbf{1}[G_i \neq G_j]\frac{\rho_{G_iG_j}}{|G_i||G_j|}, \tag{1}$$

where $\alpha_{ij} = 0$ for unmatched pairs and Z_G is the total matching weight within fragment G. The first term preserves temporal continuity within a fragment; the second spreads each nonzero cross-fragment preference uniformly over its observation pairs. This normalization assigns unit absolute weight to each non-singleton fragment and each accepted fragment comparison, preventing long fragments from dominating solely because they contain more observations. It deliberately ignores the magnitude of $|\Delta_{AB}|$ beyond the abstention threshold.

Partition objective. Let E^+ and E^- be the positively and negatively weighted observation pairs. We minimize signed disagreement over feasible partitions: $L(P) = \sum_{(i,j)\in E^+} w_{ij}\big(1 - s_{ij}(P)\big) + \sum_{(i,j)\in E^-} |w_{ij}| s_{ij}(P),\ P \in \mathfrak{P}_{\mathcal{H}}$. The first term penalizes separating positive pairs, and the second penalizes grouping negative pairs. Among partitions with equal loss, we prefer the one closest to the current grouping, measured by $D(P, P_0) = \sum_{i<j} |s_{ij}(P) - s_{ij}(P_0)|$. Thus, the resolver prioritizes agreement with the signed evidence and then minimizes unnecessary changes.

Greedy resolver and acceptance. Exact correlation clustering is NP-hard, so we use a greedy agglomerative algorithm (Keuper et al., 2015). The resolver starts from singleton observations rather than P_0, because agglomeration can merge groups but cannot split one; starting from current nodes would therefore make false merges unrecoverable. Merging groups U and V changes only their cross-pair memberships and yields gain $g(U, V) = \sum_{e_i\in U, e_j\in V} w_{ij}$. At each step, we select the feasible merge with the largest gain, where feasibility requires that no same-frame pair be merged. Positive-gain merges are accepted; zero-gain merges are accepted only when they decrease $D(P, P_0)$, with ties resolved as described in Appendix C.9.

Let $\widehat{P}$ be the resulting partition. If P_0 is feasible, we accept $\widehat{P}$ only when $L(\widehat{P}) < L(P_0)$; if P_0 violates a same-frame constraint, a feasible proposal may be accepted to restore that constraint. This gate prevents a trigger from rewriting a history that the evidence does not contradict.

For $n = |\mathcal{S}_t|$, the resolver stores at most $n(n-1)/2$ observation-pair weights and performs at most $n - 1$ merges. Rendering remains the dominant cost and scales with the number of fragment comparisons and K_{check}. The greedy resolver has no global optimality guarantee, although it is exact when the signed evidence admits a feasible zero-loss partition (Appendix C.9). It also cannot isolate an erroneous matching chain within a fragment, because the chain is never tested under separate internal hypotheses; it can be split only when cross-fragment evidence supports the cut.

3.4 Applying Revisions to Scene Graphs

A revised partition changes which observations support each node. Updating node geometry alone would leave relation edges attached to outdated identities: a split node could retain edges belonging to its sibling, while a merged node could lose edges from its duplicate. Because each relation is stored as an observation-level record, the revised assignment determines the new endpoints without rerunning predicate prediction.

Graph repair. For each revised group U, we preserve the identifier of the old node k with the largest overlap $|U \cap C_k^0|$; each old identifier is inherited at most once, and all remaining groups receive new identifiers (Appendix E). This defines the revised assignment a_t, which equals a_t^0 outside $\mathcal{S}_t$. We then remap every stored predicate record through its observation endpoints: $(e_i, r, e_j) \mapsto \big(a_t(e_i),\, r,\, a_t(e_j)\big),\ (e_i, r, e_j) \in \mathcal{R}_t$. The remapped records reaggregate node states and relation edges, reconnect edges involving revised nodes, and remove self-loop edges from the current graph. The original records remain stored for future revisions, and nodes with no supporting observations are removed. The front end then applies its own aggregation rules to the revised records; individual predicate predictions are unchanged (App. C.10).

Online execution. Revision cost grows with the size of the local history. Processing a large scope synchronously would stall the front end, queue incoming frames, and leave the output graph stale.

We therefore process regions with $|\mathcal{S}_t| \leq B_{\text{sync}}$ synchronously and submit larger regions to a background worker using a fixed snapshot. Mapping continues while the worker runs. When an accepted result is committed, observations received after the snapshot are reassigned using the revised graph. Both modes use the same evidence and resolver and differ only in when the result becomes active. During background processing, uncorrected nodes may continue to attract observations, and some of these interim assignments may remain unrevised.

4 EXPERIMENTS

We evaluate SGReviser from four perspectives: overall graph quality, correction of historical association errors, the reliability of leave-one-frame-out geometric evidence, and the trade-off between accuracy and online cost. Historical association correction and held-out evidence reliability are evaluated in App. G and App. H, respectively. Unless otherwise stated, SGReviser is attached to an existing front end without changing its detections, predictions, or association settings.

4.1 EXPERIMENTAL SETUP

Datasets. We evaluate on 3DSSG (Wald et al., 2020), built on 3RScan (Wald et al., 2019), and ReplicaSSG (Hou et al., 2025), which extends Replica (Straub et al., 2019) with semantic relations. Following FROSS (Hou et al., 2025), 3DSSG has 20 object and 7 predicate classes, while ReplicaSSG has 34 object and 9 predicate classes; we tune hyperparameters on the 3DSSG validation split and 7 of 18 ReplicaSSG scenes, using the remaining 11 for testing. We process all frames and render depth from the provided meshes with benchmark camera poses, matching the FROSS and NoPA input protocol and isolating association errors from sensor noise. Scene splits and input details are given in App. F.

Baselines. We attach SGReviser to reproduced FROSS (Hou et al., 2025) and NoPA (Yeo et al., 2026), both using the RT-DETR-EGTR detector released with FROSS (Zhao et al., 2024; Lv et al., 2024; Im et al., 2024). In paired comparisons, detections, predictions, and front-end association rules and configurations are identical; only revision changes. Published PUF (Yang et al., 2026) results provide external context, with latency reported under the original protocols.

Metrics. We report Recall@1 for relation triplets, objects, and predicates (Rel., Obj., and Pred.) and class-mean recall for objects and predicates. Predicted nodes are matched to annotated instances using the FROSS protocol; unmatched instances and missing relations count as errors. Details are provided in App. F. To measure the graph available during processing, we use streaming relation recall, $\text{sRel.} = \frac{1}{M}\sum_{\ell=1}^{M} R_{\text{Rel.}}(\tau_\ell)$, following Li et al. (2020). Here, $R_{\text{Rel}}(\tau_\ell)$ is the relation recall of the latest committed graph at time τ_ℓ. We sample every 100 ms on a 30 FPS stream and stop when the input sequence ends; subsequent background completion affects only final-graph metrics. Per-frame latency measures foreground processing time, including front-end inference, synchronous revision, and background-result write-back, but excluding queueing and parallel background computation. Unless stated otherwise, reported results are means over three runs.

Implementation details. Unless otherwise stated, ablations and additional analyses use FROSS + SGReviser on 3DSSG. Experiments use one RTX 4090 GPU. The main system processes a revision region synchronously when its size is at most B_{sync}. The default geometric, evidence, scope, and scheduling parameters and further implementation details are provided in App. E.

4.2 MAIN RESULTS

Tabs. 1 and 2 compare the front ends with and without SGReviser. On 3DSSG, relation Recall@1 improves by 16.4 and 13.0 percentage points for FROSS and NoPA, respectively. On ReplicaSSG, the corresponding gains are 4.6 and 5.7 points. Predicate mRecall also improves in all four comparisons, indicating that the gains are not limited to the most frequent predicates.

The smaller gains on ReplicaSSG are consistent with the lower object recall of its front ends, whose detector is not fine-tuned on Replica. Fewer correct observations are therefore available for revision. On 3DSSG, mean per-frame latency increases from 8.91 to 22.42 ms for FROSS and from 11.24 to 26.78 ms for NoPA. These means are below the 33.3 ms frame interval, although they do not establish a sustained real-time guarantee because tail latency and background backlog are not captured

Table 1: Results on the 3DSSG test set. † marks a reproduced front end. Subscripts give percentage-point changes relative to that front end. NR denotes not reported. Latencies of external methods follow their original protocols. The NoPA variant comparison is provided in App. F.

Method	Recall@1 (%) ↑			mRecall (%) ↑		sRel. (%) ↑	Latency (ms) ↓
	Rel.	Obj.	Pred.	Obj.	Pred.		
VGfM (Gay et al., 2018)	19.6	50.0	20.4	34.8	11.0	NR	379.00
Kim (Kim et al., 2020)	9.1	59.0	7.1	51.0	8.0	NR	454.00
SGFN (Wu et al., 2021)	22.0	51.6	27.5	37.7	24.0	NR	245.00
MonoSSG (Wu et al., 2023)	23.3	53.8	28.4	43.8	26.6	NR	283.00
SCRSSG (Yeo et al., 2025)	25.7	61.1	27.6	60.5	27.8	NR	350.00
PUF-Gaussian (Yang et al., 2026)	46.0	69.7	51.4	65.8	28.2	NR	15.00
FROSS† (Hou et al., 2025)	28.4	63.0	33.7	64.6	19.3	21.3	8.91
FROSS + Overlap	$34.5_{+6.1}$	$66.3_{+3.3}$	$40.1_{+6.4}$	$65.1_{+0.5}$	$22.7_{+3.4}$	$23.8_{+2.5}$	16.41
FROSS + SGReviser (offline)	$31.8_{+3.4}$	$64.2_{+1.2}$	$37.4_{+3.7}$	$64.9_{+0.3}$	$20.7_{+1.4}$	$21.3_{+0.0}$	12.71
FROSS + SGReviser	$\mathbf{44.8}_{+16.4}$	$\mathbf{71.5}_{+8.5}$	$\mathbf{50.4}_{+16.7}$	$\mathbf{66.3}_{+1.7}$	$\mathbf{29.9}_{+10.6}$	$\mathbf{28.1}_{+6.8}$	22.42
FROSS + GT assoc. (oracle)	$53.5_{+25.1}$	$78.2_{+15.2}$	$57.9_{+24.2}$	$72.0_{+7.4}$	$36.7_{+17.4}$	$33.4_{+12.1}$	10.02
NoPA† (Yeo et al., 2026)	53.3	69.6	61.1	66.6	29.4	18.3	11.24
NoPA + SGReviser	$\mathbf{66.3}_{+13.0}$	$\mathbf{76.7}_{+7.1}$	$\mathbf{74.1}_{+13.0}$	$\mathbf{69.3}_{+2.7}$	$\mathbf{38.6}_{+9.2}$	$\mathbf{24.3}_{+6.0}$	26.78

Table 2: Results on the ReplicaSSG test set, with the conventions of Tab. 1.

Method	Recall@1 (%) ↑			mRecall (%) ↑		sRel. (%) ↑	Latency (ms) ↓
	Rel.	Obj.	Pred.	Obj.	Pred.		
FROSS† (Hou et al., 2025)	23.3	26.5	28.8	28.9	19.6	11.0	7.47
FROSS + SGReviser	$\mathbf{27.9}_{+4.6}$	$\mathbf{32.2}_{+5.7}$	$\mathbf{36.6}_{+7.8}$	$\mathbf{31.2}_{+2.3}$	$\mathbf{27.8}_{+8.2}$	$\mathbf{15.2}_{+4.2}$	20.57
NoPA† (Yeo et al., 2026)	35.7	29.9	38.1	31.0	17.5	7.4	8.64
NoPA + SGReviser	$\mathbf{41.4}_{+5.7}$	$\mathbf{35.2}_{+5.3}$	$\mathbf{45.7}_{+7.6}$	$\mathbf{35.5}_{+4.5}$	$\mathbf{24.7}_{+7.2}$	$\mathbf{12.9}_{+5.5}$	23.03

by the mean. Despite the additional computation, sRel. improves by 4.2–6.8 points, showing that SGReviser improves the graph available during processing rather than only the final graph.

4.3 Ablation Studies

Table 3 evaluates the main components and execution settings of FROSS + SGReviser on 3DSSG. Additional sensitivity analyses for η and m, revision workload statistics, and further variants are reported in App. I.

Evidence. Removing inter-fragment evidence lowers relation Recall@1 by 11.9 points, showing that adjacent-frame matching and same-frame exclusions cannot explain the full gain. Removing mask precision ν, coverage c, or the free-space penalty ϕ also reduces final and streaming quality. Each term addresses a distinct failure: without ν, surfaces spilling outside the detection region are weakly penalized; without c, small surfaces may appear compatible despite missing observations; and without ϕ, ge-

Table 3: Ablations of FROSS + SGReviser on 3DSSG. Latency with ORB-SLAM3 poses excludes pose estimation.

Variant	Recall@1 (%) ↑			sRel. (%) ↑	Latency (ms) ↓
	Obj.	Pred.	Rel.		
Full system ($K_{\text{cand}} = 3$)	71.5	50.4	44.8	28.1	22.42
Evidence					
No inter-fragment evidence	65.7	40.0	32.9	21.5	17.37
No mask precision ν	66.4	42.1	36.2	23.2	19.26
No coverage c	67.2	42.3	35.3	22.3	20.31
No free-space penalty ϕ	67.5	43.5	34.8	24.2	19.43
Input geometry					
Ground-truth masks	73.8	55.0	49.7	30.0	23.52
ORB-SLAM3 poses	68.2	46.8	41.2	26.9	23.11
Scope and scheduling					
$K_{\text{cand}} = 1$	65.2	41.6	33.9	22.7	15.84
$K_{\text{cand}} = 5$	72.3	52.0	46.1	19.3	33.76
Synchronous only	74.3	57.7	52.9	12.5	48.57

ometry in observed free space is not penalized. Ground-truth masks improve relation Recall@1 by 4.9 points, indicating that background and neighboring-object points inside detection boxes limit revision quality.

Input geometry. Replacing benchmark poses with ORB-SLAM3 (Campos et al., 2021) estimates reduces relation Recall@1 and sRel. by 3.6 and 1.2 points, respectively. SGReviser therefore depends on reasonably accurate geometric alignment but retains most of its gain under the tested poses. Reported latency excludes pose estimation. The paired comparison is provided in App. I.

Revision scope. Reducing K_{cand} from 3 to 1 lowers relation Recall@1 by 10.9 points because the smaller scope may exclude nodes needed to repair false splits. Increasing it to 5 gains 1.3 points in final recall, but raises latency to 33.76 ms and lowers sRel. by 8.8 points. A wider scope improves the final graph at the cost of streaming quality.

Execution schedule. Processing every revision synchronously gives the highest final relation recall, but raises latency to 48.57 ms, above the 33.3 ms frame interval. The graph therefore lags the input, and sRel. falls to 12.5, below the FROSS baseline. Background execution trades some immediate correction for a more timely graph, explaining the gap between final and streaming recall.

4.4 Behavior over a Sequence

Figure 3 follows FROSS and FROSS + SGReviser through the 1,281-frame 3DSSG sequence `38770ca5`, selected because it contains many revision events. The object- and predicate-recall gaps emerge early and persist, averaging 7.5 and 12.6 points over the sequence. Synchronous revisions are concentrated in the early and middle stages, where accumulated association errors are removed. Later revisions involve longer histories and are more often processed in the background; foreground latency therefore returns closer to the FROSS baseline while recall remains higher.

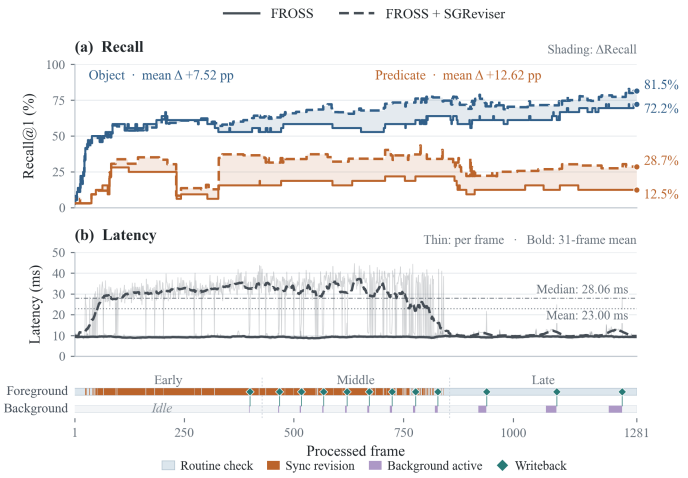

Figure 3: Graph quality and online latency on sequence `38770ca5`. (a) Object and predicate Recall@1. (b) Latency and revision events. Solid and dashed curves denote FROSS and FROSS + SGReviser.

The figure reports per-frame latency together with a 31-frame centered moving average and marks synchronous revisions, background jobs, and write-backs. Runtime breakdown, memory usage, and aggregate statistics over all test sequences are provided in App. J.2 and App. K.

5 Conclusion

We introduced ReSGG, an online setting in which historical observation-to-node assignments can be revised as new views provide evidence. SGReviser retains observation-level provenance, detects local geometric conflicts, and compares joint and separate leave-one-frame-out reconstructions. Signed geometric evidence then guides a constrained observation partition that can split false merges and join false splits. The revised partition also reconnects relation edges through stored observation endpoints, without rerunning predicate prediction. Small revisions run synchronously, while larger ones can be processed in the background without additional task-specific training. Experiments with FROSS and NoPA on 3DSSG and ReplicaSSG improve relation Recall@1 by a large margin across four comparisons. Future work should address dynamic scenes, bounded history and backlog management, and more powerful revision solvers.

AI USE STATEMENT

During manuscript preparation, generative AI tools assisted in refining the wording and improving readability. The methodology details any use of these tools as part of the research itself. All AI-assisted revisions were reviewed by the authors, who retain full responsibility for the accuracy and integrity of the final manuscript.

REFERENCES

Iro Armeni, Zhi-Yang He, JunYoung Gwak, Amir R. Zamir, Martin Fischer, Jitendra Malik, and Silvio Savarese. 3D scene graph: A structure for unified semantics, 3D space, and camera. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5664–5673, 2019.

Nikhil Bansal, Avrim Blum, and Shuchi Chawla. Correlation clustering. *Machine Learning*, 56 (1–3):89–113, 2004.

Deniz Bickici, Michael Pabst, Shohei Mori, and Dieter Schmalstieg. Think while you map: Asynchronous vision-language agents for incremental 3D scene graphs. In *Proceedings of the European Conference on Computer Vision*, pp. 411–429, 2026.

Carlos Campos, Richard Elvira, Juan J. Gómez Rodríguez, José M. M. Montiel, and Juan D. Tardós. ORB-SLAM3: An accurate open-source library for visual, visual-inertial, and multimap SLAM. *IEEE Transactions on Robotics*, 37(6):1874–1890, 2021.

Brian Curless and Marc Levoy. A volumetric method for building complex models from range images. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, pp. 303–312, 1996.

Paul Gay, Stuart James, and Alessio Del Bue. Visual graphs from motion (VGfM): Scene understanding with object geometry reasoning. In *Proceedings of the Asian Conference on Computer Vision*, pp. 330–346, 2018.

Margarita Grinvald, Fadri Furrer, Tonci Novkovic, Jen Jen Chung, Cesar Cadena, Roland Siegwart, and Juan Nieto. Volumetric instance-aware semantic mapping and 3D object discovery. *IEEE Robotics and Automation Letters*, 4(3):3037–3044, 2019.

Qiao Gu, Alihusein Kuwajerwala, Sacha Morin, Krishna Murthy Jatavallabhula, Bipasha Sen, Aditya Agarwal, Corban Rivera, William Paul, Kirsty Ellis, Rama Chellappa, Chuang Gan, Celso Miguel de Melo, Joshua B. Tenenbaum, Antonio Torralba, Florian Shkurti, and Liam Paull. ConceptGraphs: Open-vocabulary 3D scene graphs for perception and planning. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 5021–5028, 2024.

Hao-Yu Hou, Chun-Yi Lee, Motoharu Sonogashira, and Yasutomo Kawanishi. FROSS: Faster-than-real-time online 3D semantic scene graph generation from RGB-D images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 28818–28827, 2025.

Nathan Hughes, Yun Chang, and Luca Carlone. Hydra: A real-time spatial perception system for 3D scene graph construction and optimization. In *Proceedings of Robotics: Science and Systems*, 2022.

Jinbae Im, JeongYeon Nam, Nokyung Park, Hyungmin Lee, and Seunghyun Park. EGTR: Extracting graph from transformer for scene graph generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 24229–24238, 2024.

Margret Keuper, Evgeny Levinkov, Nicolas Bonneel, Guillaume Lavoué, Thomas Brox, and Bjoern Andres. Efficient decomposition of image and mesh graphs by lifted multicuts. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1751–1759, 2015.

Ue-Hwan Kim, Jin-Man Park, Taek-Jin Song, and Jong-Hwan Kim. 3-D scene graph: A sparse and semantic representation of physical environments for intelligent agents. *IEEE Transactions on Cybernetics*, 50(12):4921–4933, 2020.

Harold W. Kuhn. The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1–2):83–97, 1955.

Mengtian Li, Yu-Xiong Wang, and Deva Ramanan. Towards streaming perception. In *Proceedings of the European Conference on Computer Vision*, pp. 473–488, 2020.

Wenyu Lv, Yian Zhao, Qinyao Chang, Kui Huang, Guanzhong Wang, and Yi Liu. RT-DETRv2: Improved baseline with bag-of-freebies for real-time detection transformer. *arXiv preprint arXiv:2407.17140*, 2024.

John McCormac, Ronald Clark, Michael Bloesch, Andrew Davison, and Stefan Leutenegger. Fusion++: Volumetric object-level SLAM. In *Proceedings of the International Conference on 3D Vision*, pp. 32–41, 2018.

James Munkres. Algorithms for the assignment and transportation problems. *Journal of the Society for Industrial and Applied Mathematics*, 5(1):32–38, 1957.

Krishan Rana, Jesse Haviland, Sourav Garg, Jad Abou-Chakra, Ian Reid, and Niko Suenderhauf. SayPlan: Grounding large language models using 3D scene graphs for scalable robot task planning. In *Proceedings of the Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pp. 23–72, 2023.

Lukas Schmid, Jeffrey Delmerico, Johannes L. Schönberger, Juan Nieto, Marc Pollefeys, Roland Siegwart, and Cesar Cadena. Panoptic Multi-TSDFs: a flexible representation for online multi-resolution volumetric mapping and long-term dynamic scene consistency. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 8018–8024, 2022.

Lukas Schmid, Marcus Abate, Yun Chang, and Luca Carlone. Khronos: A unified approach for spatio-temporal metric-semantic SLAM in dynamic environments. In *Proceedings of Robotics: Science and Systems*, 2024.

Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J. Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, Anton Clarkson, Mingfei Yan, Brian Budge, Yajie Yan, Xiaqing Pan, June Yon, Yuyang Zou, Kimberly Leon, Nigel Carter, Jesus Briales, Tyler Gillingham, Elias Mueggler, Luis Pesqueira, Manolis Savva, Dhruv Batra, Hauke M. Strasdat, Renzo De Nardi, Michael Goesele, Steven Lovegrove, and Richard Newcombe. The Replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*, 2019.

Anke van Zuylen and David P. Williamson. Deterministic pivoting algorithms for constrained ranking and clustering problems. *Mathematics of Operations Research*, 34(3):594–620, 2009.

Kiri Wagstaff, Claire Cardie, Seth Rogers, and Stefan Schrödl. Constrained K-means clustering with background knowledge. In *Proceedings of the Eighteenth International Conference on Machine Learning*, pp. 577–584, 2001.

Johanna Wald, Armen Avetisyan, Nassir Navab, Federico Tombari, and Matthias Nießner. RIO: 3D object instance re-localization in changing indoor environments. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7658–7667, 2019.

Johanna Wald, Helisa Dhamo, Nassir Navab, and Federico Tombari. Learning 3D semantic scene graphs from 3D indoor reconstructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3961–3970, 2020.

Shun-Cheng Wu, Johanna Wald, Keisuke Tateno, Nassir Navab, and Federico Tombari. SceneGraphFusion: Incremental 3D scene graph prediction from RGB-D sequences. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7515–7525, 2021.

Shun-Cheng Wu, Keisuke Tateno, Nassir Navab, and Federico Tombari. Incremental 3D semantic scene graph prediction from RGB sequences. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5064–5074, 2023.

Yi Yang, Myrna Castillo, Bodo Rosenhahn, and Michael Ying Yang. PUF: Plug-and-play uncertainty-aware fusion for online 3D scene graph generation. In *Proceedings of the European Conference on Computer Vision*, pp. 462–479, 2026.

Qi Xun Yeo, Yanyan Li, and Gim Hee Lee. Statistical confidence rescoring for robust 3D scene graph generation from multi-view images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 24999–25008, 2025.

Qi Xun Yeo, Seungjun Lee, Yan Li, and Gim Hee Lee. NoPA: Non-parametric online 3D scene graph generation. In *Proceedings of the European Conference on Computer Vision*, pp. 94–112, 2026.

Hang Yin, Xiuwei Xu, Zhenyu Wu, Jie Zhou, and Jiwen Lu. SG-Nav: Online 3D scene graph prompting for LLM-based zero-shot object navigation. In *Advances in Neural Information Processing Systems*, volume 37, pp. 5285–5307, 2024.

Yian Zhao, Wenyu Lv, Shangliang Xu, Jinman Wei, Guanzhong Wang, Qingqing Dang, Yi Liu, and Jie Chen. DETRs beat YOLOs on real-time object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16965–16974, 2024.