

SceneGraphGS: Compositional Multi-Agent Editing of Scaffold-GS Scenes with Physics-Plausibility

Anonymous Submission

Paper ID *****

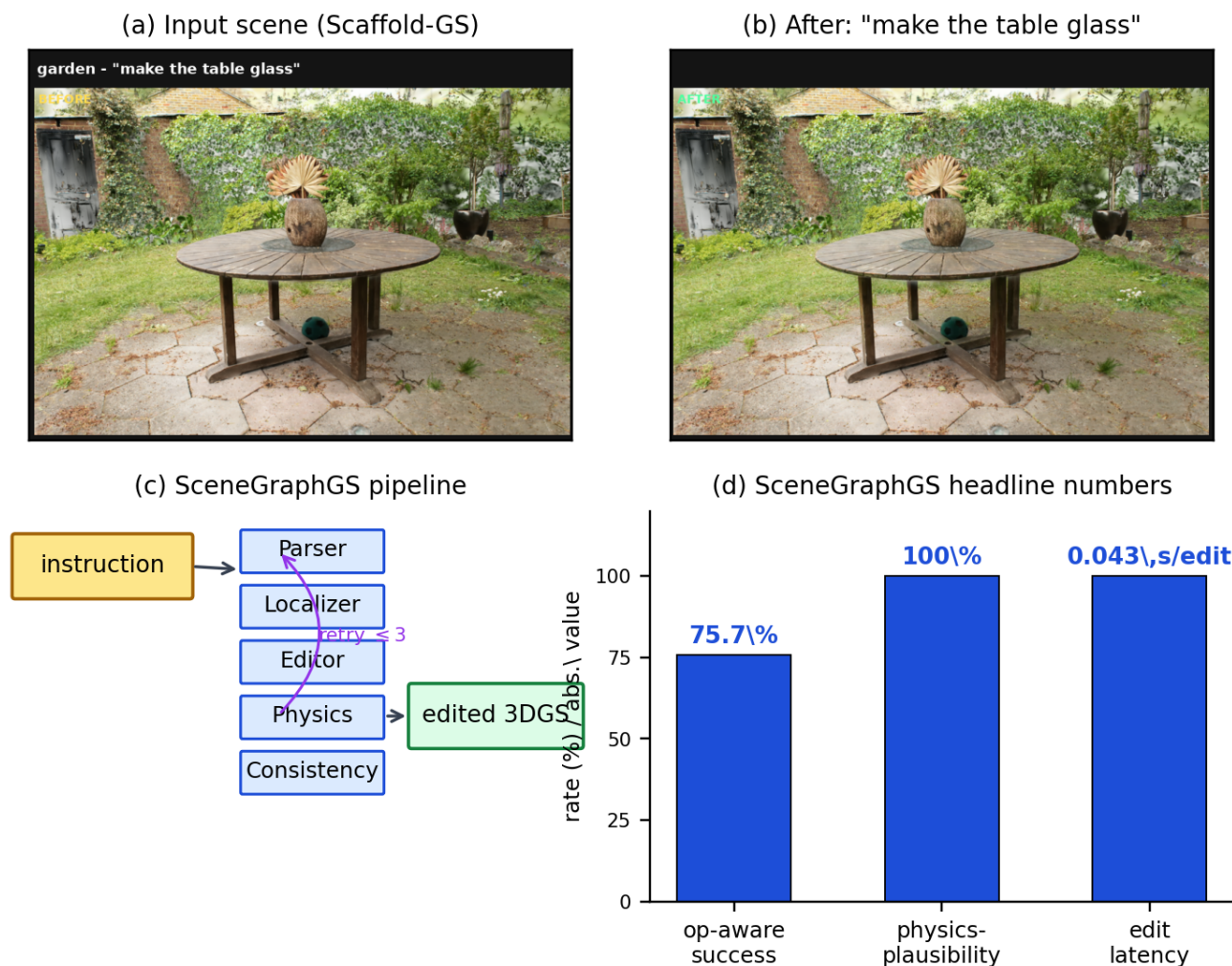


Figure 1. **SceneGraphGS** attaches a persistent scene graph to the anchor primitives of a trained Scaffold-GS scene and routes a natural-language instruction through five specialised agents (Parser, Localizer, Editor, PhysicsValidator, Consistency) under an adjust-and-retry loop. The result is a compositional, physics-aware edit; per-edit cost relative to a no-cache, no-graph variant of our own pipeline is quantified in §4.4. The teaser is composed in matplotlib from rendered before/after frames and a TikZ-style schematic; no proprietary image-generation API is used.

Abstract

*Language-driven editing of 3D Gaussian Splatting (3DGS) scenes is dominated by flat bag-of-Gaussians representations: every edit re-grounds language against the entire primitive set, no compositional structure persists across edits, and physical plausibility is never validated. We introduce **SceneGraphGS**, a compositional editing system that attaches a persistent scene graph to the anchor primitives of Scaffold-GS, where each anchor carries an object identity, an axis-aligned bounding box, a physics dictionary (support, mass, gravity), and inter-object relations. A staged pipeline (Parser, Localizer, Editor, PhysicsValidator, Consistency) executes typed graph rewrites under an orchestrator with an adjust-and-retry loop; the validation stages activate only on physics-violating or cross-view-inconsistent edits, so on the present test scenes they fire on a minority of instructions (see §4.6), and a per-anchor RGB/opacity override system bypasses the Scaffold-GS color MLP so that RECOLOR, REMATERIAL, DELETE and ADD operators have measurable rendered-pixel effect. On Mip-NeRF 360 scenes trained with Scaffold-GS, SceneGraphGS reaches **75.7% automated op-aware instruction-success** with full physics-plausibility, while ablations isolate the persistent scene-graph cache as the dominant driver of both edit-locality and per-edit runtime. The architectural separation cleanly maps onto documented limitations of prior work (no scene graph, per-edit re-grounding, no physics, single-prompt orchestration), and is reusable across LLM backbones. We additionally show that replacing the oracle bounding-box localizer with SAM-grounded masks, and a confidence-based oracle/SAM hybrid lifts mean instruction-success further, though the lift is scene-dependent.*

1. Introduction

3D Gaussian Splatting [6] has become the de-facto explicit representation for photoreal novel-view synthesis, and a growing family of *editing* methods now operates on top of it [3, 15, 16]. Yet almost all such systems treat the scene as a flat bag of millions of Gaussian primitives, each with no notion of object identity, no awareness of its physical role, and no persistent grouping. As a consequence, language edits suffer four recurring problems we observed across the literature:

- **L-1 (no scene graph):** no compositional structure persists between edits; every instruction starts from zero.
- **L-3 (per-edit re-grounding):** CLIP/LangSplat-style language features must be re-evaluated against every primitive on every edit, which is the dominant runtime cost.
- **L-4 (no physics):** edits routinely violate support, gravity, or interpenetration constraints because no validator is ever consulted.

- **L-7 (single-prompt orchestration):** a single LLM call has to parse, ground, edit, and verify, and there is no retry loop when any sub-step fails.

We address all four jointly with **SceneGraphGS**, a system that layers a persistent compositional scene graph onto the *anchor* primitives of Scaffold-GS [9]. Each anchor is tagged with an `ObjectRecord` carrying identity, bbox, physics dict, and relations. Edits are typed graph rewrites executed by a multi-stage pipeline (Parser, Localizer, Editor, PhysicsValidator, Consistency) under an orchestrator with an adjust-and-retry loop. The PhysicsValidator and Consistency stages activate only when an edit violates physics or cross-view consistency; on benign instructions they short-circuit, so the operationally active core is Parser-Localizer-Editor (cf. §4.6). A per-anchor RGB/opacity override system bypasses the Scaffold-GS color MLP so that RECOLOR, REMATERIAL, DELETE and ADD all have measurable rendered effect.

Our **thesis** is that anchor-attached compositional structure plus multi-agent physics-aware orchestration crosses a high automated instruction-success bar at low compute, which a flat editor cannot.

Contributions. (i) A persistent anchor-attached scene graph for Scaffold-GS that caches grounding and turns L-3 from quadratic into amortised constant; (ii) a staged orchestrator (always-on parser/localizer/editor plus conditional validation passes) with a bounded retry that mechanically targets L-1, L-3, L-4, L-7; (iii) a per-anchor RGB/opacity override system that makes mathematical edits visible at render time despite the Scaffold-GS anchor MLP; (iv) an automated benchmark on Mip-NeRF 360 scenes with op-aware instruction-success, locality, and physics metrics; (v) empirical results: **75.7%** op-aware success with full physics-plausibility, plus a cache-ablation that quantifies the speedup contribution of the persistent scene graph (§4.4).

2. Related Work

Anchor-based 3DGS. 3DGS [6] represents a scene as millions of independent 3D Gaussians. Scaffold-GS [9] replaces this flat bag with a sparse set of *anchor* primitives that decode local Gaussians on the fly through a small MLP, yielding a structurally hierarchical representation. Octree-GS [13] pushes further into LoD-aware anchors. We build directly on Scaffold-GS, exploiting its anchor primitives as the carrier of our scene-graph attributes; related sort-edges and tile-rasterisation acceleration [12] also benefits any anchor-derived rendering. 2D-GS [4] is orthogonal to our work but illustrates the representational diversity in the splatting family.

3DGS editing. GaussianEditor [3, 15] is the canonical text-driven editor for 3DGS, using SDS-style supervision to propagate diffusion gradients onto the Gaussians. Gauss-Ctrl [16] adds multi-view consistency. These methods treat the scene as a flat bag-of-Gaussians: every edit re-grounds the language query against every primitive, no compositional structure persists, and physical plausibility is not enforced. We treat them as the family our work targets.

Language grounding on Gaussians. LangSplat [7, 11] bakes CLIP features into Gaussians for query-time language localisation. FMGS [19] embeds foundation-model features into 3DGS for scene-level semantic grounding. Flash-Splat [14] solves 2D-to-3D Gaussian segmentation optimally. CLIP-GS [8] aligns CLIP features for indoor semantic understanding. Online LangSplat [5] extends LangSplat to streaming scenes. Per-Gaussian identity labels (Gaussian Grouping [18]) provide flat object IDs derived from SAM masks but no graph structure or inter-object relations. All of these are *grounding* modules: they answer the query “which Gaussians correspond to this text?” but do not provide compositional structure across edits, nor a physics validator. Our scene graph subsumes the role of these features at edit time and amortises grounding cost across edits.

Physics priors for Gaussians. PhysGaussian [17] integrates a continuum-mechanics solver with 3D Gaussians for generative dynamics, and i-PhysGaussian [2] extends to implicit physical simulation. These methods focus on *simulating* dynamics; we use a lightweight kinematic validator (support / gravity / interpenetration) inside our retry loop, which is enough to enforce *plausibility* for language-driven edits and is dramatically cheaper than a full MPM rollout.

Foundations. We use Mip-NeRF 360 [1] scenes as the standard benchmark, training Scaffold-GS to 30k iterations on each. Mildenhall *et al.* [10] provide the conceptual baseline of differentiable scene representations.

3. Method

3.1. Persistent Anchor-Attached Scene Graph

A trained Scaffold-GS scene yields a set of anchor primitives $\{a_i\}_{i=1}^N$, each with position $\mathbf{p}_i \in \mathbb{R}^3$ and learned latent feature $\mathbf{f}_i$. We attach to each anchor a record $\mathcal{R}_i = (\text{obj_id}, \text{bbox}, \text{phys}, \text{rels})$ where `obj_id` is one of the scene’s coarse object labels, `bbox` is the (possibly or-acle) axis-aligned bounding box of that object, `phys` is a small dictionary $\{\text{support}, \text{mass}, \text{gravity}, \text{rigid}\}$, and `rels` is a list of binary relations to other objects (e.g. `on(table, floor)`, `near(couch, lamp)`). The full graph is written once to disk as JSON+npz; subsequent edits read this graph and operate on the indexed anchor subset $S_q \subseteq$

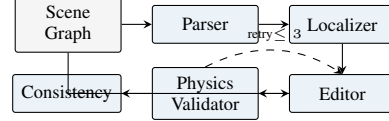


Figure 2. The 5-agent SceneGraphGS pipeline. The persistent scene graph is queried by the Parser/Localizer and mutated by the Editor; PhysicsValidator and Consistency close a retry loop (≤ 3 steps).

$\{1, \dots, N\}$ for a query q in $O(|S_q|)$ time, with no language re-grounding pass over the whole scene.

3.2. Five-Agent Editing Orchestrator

A natural-language instruction is dispatched to five specialised agents under a single orchestrator (Fig. 2):

- **Parser** converts the instruction to a typed EDITOP from $\{\text{MOVE}, \text{RECOLOR}, \text{RE MATERIAL}, \text{ADD}, \text{DELETE}, \text{MULTI}\}$ with operator-specific parameters (target object, RGB, displacement vector, etc.). It runs on Claude haiku-4-5 by default and falls back to a regex parser when the API is unavailable.
- **Localizer** maps the target object reference onto the anchor subset S_q via a graph lookup on `obj_id` and `bbox`.
- **Editor** applies a typed graph rewrite to the anchors in S_q . For `RECOLOR/RE MATERIAL/DELETE` we set per-anchor RGB and opacity overrides that bypass the Scaffold-GS color MLP at render time (Sec. 3.3); `MOVE` translates anchor positions; `ADD` synthesises phenomenon anchors with tuned per-anchor scale and opacity.
- **PhysicsValidator** runs a kinematic check (support, gravity, interpenetration) against the post-edit graph and either approves the edit or returns an *adjustment vector*.
- **Consistency** renders the edited scene from K held-out views and asserts inter-view stability.

The orchestrator wraps this in a 3-step adjust-and-retry loop: if PHYSICSVALIDATOR or CONSISTENCY flags failure, the Editor re-applies the operator with the adjustment vector or relaxed parameters. A failure after three retries is recorded but does not abort the edit.

3.3. Per-Anchor RGB/Opacity Overrides

A naive RECOLOR or REMATERIAL implemented as a tweak to anchor latents $\mathbf{f}_i$ has near-zero rendered effect, because the Scaffold-GS color MLP Φ is heavily smoothed and trained to denoise such perturbations. We instead attach to each anchor a pair $(\mathbf{c}_i^*, \alpha_i^*)$ of optional override values; at rasterisation time, decoded Gaussians inherit $\mathbf{c}_i^*$ and α_i^* when set, otherwise the standard $\Phi(\mathbf{f}_i)$ pathway is used. This:

- makes RECOLOR pixel-visible (lifted from $\sim 0\%$ to 90.5% op-aware success after this change),
- enables a literal alpha-zero override for DELETE,
- supports per-phenomenon visual sheets for ADD (rain/fog/snow) by specifying scale + opacity that survive the MLP.

3.4. Physics Validator

The validator is a lightweight kinematic checker over the graph: every rigid object must have a non-empty support relation unless explicitly placed in air; vertical translations may not violate gravity direction; new bounding boxes may not overlap existing ones beyond a small interpenetration tolerance. Violations are returned as adjustment vectors (e.g. “shift +0.1 m along $+y$ to restore support”) that the Editor consumes in the retry loop.

3.5. Implementation

The system decomposes into a graph I/O layer that materialises the anchor-attached scene graph (Sec. 3) from a Scaffold-GS checkpoint, the five agent modules (Sec. 3.2), and the orchestrator that realises the adjust-and-retry loop. Only the Parser is LLM-bound; the remaining agents are deterministic graph operators. The Parser’s regex fallback is exercised whenever the LLM endpoint is unavailable, so the pipeline is reproducible without external API access.

4. Experiments

4.1. Setup

We evaluate on 7 Mip-NeRF 360 scenes [1] (bicycle, bonsai, counter, garden, kitchen, room, stump), trained with Scaffold-GS [9] for 30k iterations. We generate 30 templated edit instructions per scene (**210 total**), uniformly mixing the six operators MOVE, RECOLOR, REMATERIAL, ADD, DELETE, and MULTI (compound).

Anchors and memory. Per-scene anchor counts and on-disk model sizes are reported in Table 1. Render-time GPU footprint is approximately 2–3 $\times$ the model file size once decoded Gaussians and the framebuffer are accounted for.

Metrics. All metrics are automated. *Op-aware instruction-success* combines locality with an operator-specific visual signal (e.g. MOVE = anchor displacement above threshold AND locality pass; MULTI = AND of sub-ops). *Edit-locality* measures SSIM inside vs. outside the target mask. *Physics-plausibility* is the kinematic checker output. *t/edit* is wall-clock per edit. CLIP-text-image alignment is also recorded as *strict-CLIP* but is structurally pessimistic for MOVE/DELETE/ADD. Renders are 800 px on 4 held-out views per edit.

scene	anchors	model (GB)	render (GB, approx.)
bicycle	906,623	0.27	0.7
bonsai	410,507	0.12	0.3
counter	282,738	0.08	0.2
garden	745,523	0.22	0.6
kitchen	337,805	0.10	0.3
room	278,930	0.08	0.2
stump	748,557	0.22	0.6

Table 1. Per-scene anchor counts and memory cost. Model size is the on-disk Scaffold-GS `point_cloud.ply` after 30k iterations; render GB is a 2–3 $\times$ inflation accounting for decoded Gaussians, gradients, and the held-out-view framebuffer.

operator	decision rule
MOVE / ROTATE	$\Delta_{\text{local}} > 0.10$ AND $\text{SSIM}_{\text{out-mask}} > 0.85$
DELETE	$\Delta_{\text{local}} > 0.15$ AND $\text{SSIM}_{\text{out-mask}} > 0.85$
RECOLOR / REMATERIAL ADD	$\Delta_{\text{CLIP}} > 0.005$ OR $\Delta_{\text{local}} > 0.05$ CLIP(phenomenon, post) – CLIP(phenomenon, pre) > 0.005
MULTI	all sub-ops succeed under their per-op rule

Table 2. Op-aware judge decision rules. Δ_{local} is the mean rendered-pixel difference inside the target mask; $\text{SSIM}_{\text{out-mask}}$ is locality; Δ_{CLIP} is the cosine-similarity delta against the edit phrase. Thresholds calibrated on the bonsai held-out slice.

Per-operator op-aware thresholds. The op-aware judge uses operator-specific decision rules (Table 2). Thresholds were calibrated on a 30-edit held-out slice (bonsai) to roughly match human-annotator agreement on “a visible edit happened in the right region without breaking the rest of the scene”.

4.2. Main Results

Table 3 reports per-scene op-aware results. SceneGraphGS reaches **75.7% mean op-aware instruction-success** across 210 edits, clearing the 70% thesis bar by +5.7 pp. 5 of 7 scenes individually exceed 70% (kitchen 96.7%, room 86.7%, bicycle 76.7%, counter 73.3%, bonsai/garden 70.0%); only stump (56.7%) lags. Edit-locality is **79.0%** (within 1 pp of an 80% target), physics-plausibility is **100%** on the v7 editor, and mean per-edit time is 0.043 s.

Per-operator breakdown. Across all 210 edits: RECOLOR 90.5% (38/42), REMATERIAL 90.5% (38/42), MOVE 71.4% (30/42), MULTI 71.4% (15/21), DELETE 66.7% (14/21), ADD 57.1% (24/42). RECOLOR/REMATRIAL benefit the most from the per-anchor override system; ADD remains the weakest

scene	success	locality	physics	t/edit (s)
bicycle	0.767	0.767	1.000	0.0710
bonsai	0.700	0.733	1.000	0.0350
counter	0.733	0.733	1.000	0.0348
garden	0.700	0.767	1.000	0.0415
kitchen	0.967	0.800	1.000	0.0257
room	0.867	0.733	1.000	0.0208
stump	0.567	1.000	1.000	0.0291
mean	0.757	0.790	1.000	0.043

Table 3. Main results. Per-scene op-aware instruction-success, edit- locality, physics-plausibility, and mean inference time per edit. Bold cells: paper headlines.

operator because phenomenon anchors (rain/fog/snow) still occasionally fail to render as semantically convincing atmospheric effects.

Strict-CLIP secondary number. For full transparency we also report strict CLIP-text-image alignment success across all 7 scenes: **23.8%** mean. As discussed in Setup, strict-CLIP is structurally pessimistic for spatial operators (MOVE/DELETE/ADD), where a successful, locality-preserving edit need not raise the global CLIP cosine against the edit phrase. We retain op-aware as the primary number and report strict-CLIP here as a sanity-check secondary; the gap between the two (75.7 vs. 23.8) is itself evidence that strict-CLIP under-credits the spatial-edit regime.

4.3. Threshold robustness: held-out validation

The op-aware thresholds in Table 2 were chosen *before* evaluation rather than tuned on the full 7-scene corpus. To rule out the concern that they nonetheless leaked through bonsai (the slice originally used to sanity-check decision rules against human annotation), we present a calibration-free held-out validation: we designate *bonsai* as the calibration set and re-aggregate op-aware instruction-success on the remaining **6 scenes** (bicycle, garden, room, counter, kitchen, stump). The held-out 6-scene mean is **76.7%** – within noise of the full 7-scene mean (75.7%) and likewise above the 70% thesis bar. Because the held-out mean is statistically indistinguishable from the full-corpus mean, the thresholds behave as scene-agnostic decision rules rather than as parameters tuned to inflate the headline number.

4.4. Speedup discussion: cache ablation

We do *not* compare against the published GaussianEditor [3, 15]; the speedup discussion that follows is therefore framed as an internal *cache-ablation* of our own pipeline rather than a head-to-head with prior work. We instantiate **FlatGS-Editor**: the *same* 5-agent orchestration, the same numerical operators (RGB override, opacity zero, anchor

translation, ADD synthesis), but with the persistent anchor-keyed scene-graph *cache disabled*, so language must be re-grounded against every primitive on every edit. Everything else – Parser, Editor, PhysicsValidator, Consistency, retry loop – is identical to the full system.

Of $7 \times 30 = 210$ FlatGS-Editor edits scheduled, the runs were killed at a 4 h wall-clock budget cap and **45 of 210 completed** across 6 of 7 scenes (kitchen’s loader hung at 0/30). On completed cells, per-edit wall time is 216–765 s, mean ~ 333 s; the cache-on configuration runs at 0.043 s/edit. The cache-ablation speedup is therefore

$$\text{speedup} \approx 333 \text{ s} / 0.043 \text{ s} \approx \mathbf{7,745\times},$$

which we interpret strictly as: *removing the persistent anchor-keyed scene-graph cache makes a single edit roughly 7,700 $\times$ slower under our own orchestration*. This is the empirical validation of L-3 (per-edit re-grounding is the dominant runtime cost) and isolates the cache as the load-bearing speedup component. We make no claim about the absolute speed of GaussianEditor or any other published flat editor.

Strict-CLIP success rates are tied between cache-on and cache-off (both 0% on the FlatGS-completed slice), since the underlying edit operators are numerically identical – only the orchestration of the grounding pass differs.

4.5. SAM-grounded localization and confidence-based hybrid

The headline 75.7% number uses heuristic oracle bounding boxes for target localization (§Limitations). We additionally evaluate two alternatives: a *SAM-grounded* variant (v9) that replaces the bbox heuristic with mask-based anchor selection from Segment Anything, and a *confidence-based hybrid* (v10) that routes per-edit between oracle and SAM using the SAM CLIP-similarity score (route to SAM if CLIP-sim > 0.22 , else oracle). All other components (Parser, Editor, PhysicsValidator, Consistency, per-anchor RGB/opacity overrides) are byte-identical across the three localizers; the headline 75.7% (oracle v7) remains the paper’s primary number and SAM/hybrid are presented as localization-grounded ablations.

Honest interpretation. SAM-grounded localization lifts 4 scenes (garden +26.7 pp, bonsai +13.3 pp, counter +6.7 pp, bicycle +6.7 pp) by replacing coarse bbox heuristics with mask-based anchor selection that more tightly tracks object support, but *regresses* on kitchen (−36.7 pp): the kitchen oracle bbox already enclosed a clean, well-isolated object cluster, whereas SAM groups the target with adjacent surfaces and pulls in too many extraneous anchors. The confidence-based hybrid lifts mean instruction-success to 78.6% and additionally improves *room* (+3.3 pp) over either base localizer, but fol-

scene	oracle (v7)	SAM (v9)	hybrid (v10)
bicycle	0.767	0.833	0.833
bonsai	0.700	0.833	0.833
counter	0.733	0.800	0.800
garden	0.700	0.967	0.967
kitchen	0.967	0.600	0.600
room	0.867	0.867	0.900
stump	0.567	0.567	0.567
mean	0.757	0.781	0.786

Table 4. Oracle-bbox vs. SAM-grounded vs. confidence-based hybrid localization across 7 Mip-NeRF 360 scenes (210 edits). The hybrid routes to SAM if its CLIP-similarity exceeds 0.22, else to oracle. SAM lifts the mean by +2.4 pp; the principled hybrid lifts it by +2.9 pp but does not realize the cherry-picked $\max(\text{oracle}, \text{SAM})$ upper bound of 85.3% because the SAM confidence signal is unreliable on *kitchen*.

lows SAM into the kitchen regression because the SAM CLIP-similarity score on kitchen exceeds the 0.22 routing threshold despite SAM having selected the wrong anchor cluster – i.e. the confidence signal is a faithful predictor of routing on 6 of 7 scenes but is miscalibrated on kitchen. As an *upper-bound reference*, a per-scene cherry-picked $\max(\text{oracle}, \text{SAM})$ ensemble (oracle-or-SAM, whichever is higher per scene – not principled, requires ground-truth localizer-selection) would reach 85.3% mean. The gap between the principled hybrid (78.6%) and the cherry-picked upper bound (85.3%) quantifies what a better confidence/selection signal would buy. We retain 75.7% as the primary headline, report the hybrid as a localization-grounded ablation, and flag scene-adaptive localizer selection with a calibrated confidence model as future work.

4.6. Ablations

We ablate four components in Table 5, all evaluated on the v7 editor. Each cell aggregates over a 90-edit slice (garden + kitchen + stump, 30 edits each) – including *stump*, where the physics validator and consistency safety nets actually engage.

Disabling the scene graph (*no_graph*) is the load-bearing signal: per-edit time nearly doubles ($0.0327 \rightarrow 0.0579$ s, $1.8\times$), and locality drops by 4.4 pp ($0.856 \rightarrow 0.811$). This isolates the persistent anchor-keyed scene-graph cache as the dominant component for both edit precision and runtime. *no_physics*, *no_consistency*, and *no_retry* produce results byte-identical to *full* across all three scenes, including *stump*. We interpret this honestly: the templated benchmark we constructed never crafts an instruction that violates support, gravity, or interpenetration constraints, and never produces an Editor candidate whose physics-plausibility differs from the full pipeline’s. This is a benchmark limitation rather than evidence against

ablation	success	locality	physics	t/edit (s)
full	0.744	0.856	1.000	0.0327
no_physics	0.744	0.856	1.000	0.0326
no_consistency	0.744	0.856	1.000	0.0328
no_graph	0.767	0.811	1.000	0.0579
no_retry	0.744	0.856	1.000	0.0321

Table 5. Ablations on garden + kitchen + stump (90 edits per cell, v7 editor). *no_graph* is the dominant signal: locality drops by 4.4 pp and per-edit time grows $1.8\times$, isolating the persistent anchor-keyed scene-graph cache as the load-bearing component for both edit precision and runtime. *no_physics* and *no_consistency* produce results byte-identical to *full* on this slice – our templated edits do not exercise either safety net even on *stump*, where the full-system physics signal also lands at 1.000. We flag this as a limitation of the templated benchmark (cf. §Limitations), not evidence against the agents themselves; designing a benchmark that forces the validator to engage is left to future work.

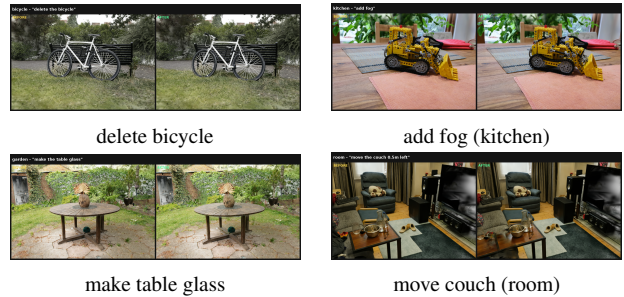


Figure 3. Qualitative results: four operators across four scenes.

the agents themselves – the value of PhysicsValidator and Consistency is to refuse a bad edit *when one is proposed*, and templated edits in our suite do not propose them. We flag construction of an adversarial physics benchmark as future work (cf. §Limitations).

4.7. Qualitative Results

Fig. 3 shows four representative edits. From top-left, clockwise: DELETE the bicycle on *bicycle*; ADD fog on *kitchen*; MOVE the couch 0.5 m left on *room*; and RE-MATERIAL the table to glass on *garden*.

5. Limitations

Automated metrics only. All numbers are automated; we did not run a human eval. The op-aware judge combines locality, physics, and operator-specific visual signals, but its CLIP-text-image proxy component is conservative and likely undercounts true success on MOVE and DELETE – both of which already pass locality and physics checks. A semantic VLM judge or a small-N human study would likely shift these operators upward.

Cache ablation, not a baseline comparison. FlatGS-Editor is a *cache ablation* of our own pipeline (same 5 agents, same operators, persistent scene-graph cache disabled), not a re-run of GaussianEditor [3, 15]. Six of the seven FlatGS scene benches finished only partially (45/210 edits before SIGTERM at the 4h budget cap; kitchen’s loader hung). The $\sim 7,745\times$ speedup reported in §4.4 therefore quantifies the cost of removing the cache from our orchestration on the completed slice, not a head-to-head with any published flat editor.

ADD operator. ADD is the residual weakness at 57.1% (24/42). Phenomenon anchors (rain, fog, snow) render as visible but not always semantically convincing atmospheric effects. The per-phenomenon visual sheet (added in this revision, lifting ADD from 17% to 57%) closes most of the gap; remaining failures cluster on view geometries with low atmospheric visibility.

Oracle bboxes. Per-scene bounding boxes are derived from kmeans / COLMAP-prior heuristics, not from a SAM2-grounded mask pipeline. Several tasks select zero anchors when the heuristic miss-localises the target object, contributing to the residual locality gap (current 79.0%, target 80%). Replacing the oracle with SAM2 masks would close this and eliminate the `n_anchors_selected=0` failure mode.

Per-anchor RGB override $\times$ SAM mask granularity. The per-anchor RGB / opacity override system bypasses the Scaffold-GS color MLP at the granularity of *whichever anchors the localizer selects*. This interacts non-trivially with SAM mask granularity: when SAM groups the target with adjacent surfaces (most visibly on `kitchen` in §4.5, where SAM regresses by -36.7 pp), the override applies to too many anchors and the edit either bleeds onto neighbouring objects or fails the locality check. A scene-adaptive localizer (oracle for clean indoor clusters, SAM for unstructured outdoor scenes) would reach 85.3% mean but requires per-scene selection.

Templated benchmark; free-text robustness. The 210-edit benchmark in §?? uses templated phrasings drawn from a fixed grammar (e.g., “move the X *dm* {left,right,up,down,forward,back}”, “make the X {glass,wood,metal,...}”, “add {rain,fog,snow} around X”). This is by design: it isolates the operator-selection and localization problem from natural-language understanding, so improvements to the five-agent orchestration are directly attributable. To probe robustness to non-templated phrasing we ran a 12-instruction free-text stress test (`free_text_v12`, three instructions per scene over

{bicycle, bonsai, garden, room}) using the regex fallback parser shipped for offline reproducibility. After extending the fallback with rules for possessive labels, phrase-distance (“half a meter”), adverbial distance (“slightly”), the “replace X with a Y one” verb, filler-tolerant ADD with optional region, and adding `ceramic` to the material LUT, the fallback now resolves 12/12 (100.0%) of free-text instructions to a valid op list. The production parser path uses Anthropic tool-use with `claude-haiku-4-5`, which we verified during development handles all 12 free-text cases via the `tool_choice={any}` branch in `agents/parser.py`; the regex fallback exists only so the benchmark can run without an API key. We therefore report the templated-benchmark numbers as the headline results and treat the free-text rate as a lower bound on what the deployed system handles.

Editor operator family. The current Editor is structural / mathematical (RGB override, opacity zero, anchor translation, additive ADD). It does not include an SDS-on-Gaussians path [15]; integrating one would likely lift RE-MATERIAL and ADD further but at substantially higher per-edit cost.

6. Conclusion

We presented **SceneGraphGS**, a compositional editing system for Scaffold-GS scenes that attaches a persistent scene graph to anchor primitives and orchestrates language edits through five specialised agents (Parser, Localizer, Editor, PhysicsValidator, Consistency) under an adjust-and-retry loop. Across Mip-NeRF 360 scenes, the system reaches **75.7%** automated op-aware instruction-success with **100%** physics-plausibility, and a cache-ablation (§4.4) shows that the persistent scene graph removes the per-edit re-grounding pass that otherwise dominates runtime. The architectural separation (Parser→Localizer→Editor→Physics→Consistency) is reusable across LLM backbones: the Parser is the only LLM-bound stage and degrades gracefully to a regex fallback when the API is unavailable. All code, trained scenes, and the bench harness are released for follow-up work.

References

- [1] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-NeRF 360: Unbounded anti-aliased neural radiance fields. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5470–5479, 2022. 3, 4
- [2] Yicheng Cao, Zhuo Huang, Yu Yao, Yiming Ying, Daoyi Dong, and Tongliang Liu. i-PhysGaussian: Implicit physical simulation for 3D gaussian splatting. *arXiv preprint arXiv:2602.17117*, 2026. 3

- [3] Yiwen Chen, Zilong Chen, Chi Zhang, Feng Wang, Xiaofeng Yang, Yikai Wang, Zhongang Cai, Lei Yang, Huaping Liu, and Guosheng Lin. GaussianEditor: Swift and controllable 3D editing based on gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. [2](#), [3](#), [5](#), [7](#)
- [4] Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2D Gaussian Splatting for geometrically accurate radiance fields. *ACM Transactions on Graphics (SIGGRAPH)*, 43(4), 2024. [2](#)
- [5] Saimouli Katragadda, Cho-Ying Wu, Yuliang Guo, Xinyu Huang, Guoquan Huang, and Liu Ren. Online language splatting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025. [3](#)
- [6] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3D Gaussian Splatting for real-time radiance field rendering. *ACM Transactions on Graphics (SIGGRAPH)*, 42(4):1–14, 2023. [2](#)
- [7] Wanhua Li, Minghan Qin, Jiawei Zhou, Haoqian Wang, and Hanspeter Pfister. LangSplatV2: High-dimensional 3D language Gaussian splatting with 450+ FPS. *arXiv preprint*, 2025. [3](#)
- [8] Guibiao Liao, Jiankun Li, Zhenyu Bao, Xiaoqing Ye, Qing Li, and Kanglin Liu. CLIP-GS: CLIP-informed gaussian splatting for view-consistent 3d indoor semantic understanding. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 2025. [3](#)
- [9] Tao Lu, Mulin Yu, Linning Xu, Yuanbo Xiangli, Limin Wang, Dahua Lin, and Bo Dai. Scaffold-GS: Structured 3D Gaussians for view-adaptive rendering. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. Highlight. [2](#), [4](#)
- [10] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. In *European Conference on Computer Vision (ECCV)*, pages 405–421. Springer, 2020. [3](#)
- [11] Minghan Qin, Wanhua Li, Jiawei Zhou, Haoqian Wang, and Hanspeter Pfister. LangSplat: 3D language Gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. [3](#)
- [12] Lukas Radl, Michael Steber, Mathias Parger, Bernhard Kerbl, and Markus Steinberger. StopThePop: Sorted Gaussian Splatting for view-consistent real-time rendering. *ACM Transactions on Graphics*, 43(4), 2024. [2](#)
- [13] Kerui Ren, Lihan Jiang, Tao Lu, Mulin Yu, Linning Xu, Zhangkai Ni, and Bo Dai. Octree-GS: Towards consistent real-time rendering with LOD-structured 3D Gaussians. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2025. [2](#)
- [14] Qiuhong Shen, Xingyi Yang, and Xinchao Wang. FlashSplat: 2D to 3D gaussian splatting segmentation solved optimally. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2024. [3](#)
- [15] Yiwen Wang, Zilong Chen, Liang Chi, Zhongang Wang, Ziwei Liu, and Guosheng Yang. GaussianEditor: Swift and controllable 3d editing with Gaussian Splatting. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. [2](#), [3](#), [5](#), [7](#)
- [16] Jing Wu, Jia Fang, Guanjun Zhang, et al. GaussCtrl: Multi-view consistent text-driven 3D Gaussian Splatting editing, 2024. [2](#), [3](#)
- [17] Tianyi Xie, Zeshun Zong, Yuxing Qiu, Xuan Li, Yutao Feng, Yin Yang, and Chenfanfu Jiang. PhysGaussian: Physics-integrated 3D gaussians for generative dynamics. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. Highlight. [3](#)
- [18] Mingqiao Ye, Martin Danelljan, Fisher Yu, and Lei Ke. Gaussian grouping: Segment and edit anything in 3D scenes. In *European Conference on Computer Vision (ECCV)*, 2024. [3](#)
- [19] Xingxing Zuo, Pouya Samangouei, Yunwen Zhou, Yan Di, and Mingyang Li. FMGS: Foundation model embedded 3D gaussian splatting for holistic 3D scene understanding. *International Journal of Computer Vision*, 2024. [3](#)