# RSI-LGS: RECURSIVE SELF-IMPROVEMENT FOR LANGUAGE GAUSSIAN SPLATTING

**Anonymous authors**
Paper under double-blind review

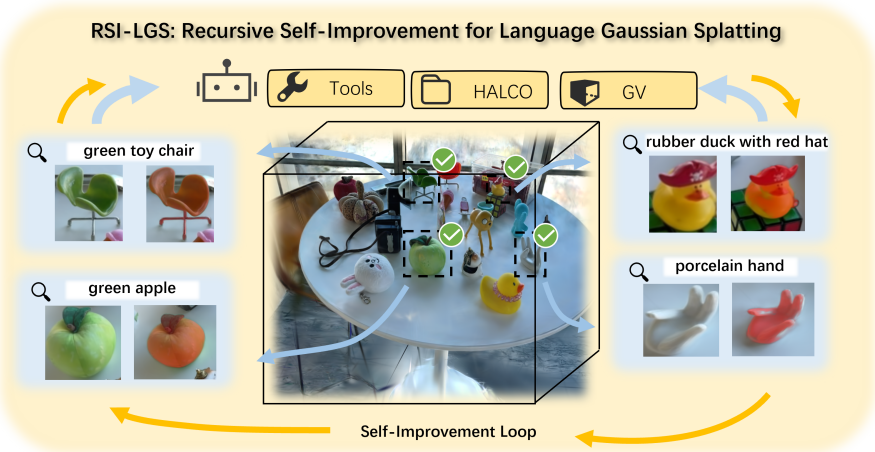



Figure 1: RSI-LGS refines an open-vocabulary 3D query over successive iterations of the self-improvement loop. Each round treats the current prediction as feedback rather than as the final result, reviews it against the rendered evidence, and adjusts the pipeline before querying again, so the prediction improves as the loop proceeds.

## ABSTRACT

Open-vocabulary 3D querying aims to localize and identify objects in 3D scenes according to arbitrary natural language prompts. Existing methods typically adopt a one-stage pipeline that aligns 3D scene representations with vision-language semantics in a static manner. Consequently, these models lack effective mechanisms for diagnosing and rectifying erroneous predictions. To address this limitation, we propose RSI-LGS, an open-vocabulary 3D querying framework equipped with a pipeline-level self-improvement loop. RSI-LGS reviews intermediate query results to diagnose localization and semantic matching errors, and adaptively adjusts the querying pipeline for subsequent iterations. This enables recursive refinement of the inference strategy and improves 3D open-vocabulary query results.

## 1 INTRODUCTION

In recent years, the rapid advancement of 3D scene representation and reconstruction technologies (Mildenhall et al., 2020; Kerbl et al., 2023) has driven 3D scene understanding beyond conventional geometric perception toward open-domain semantic understanding. Traditional approaches to 3D scene understanding typically rely on a predefined and closed set of semantic categories (Zhi et al., 2021), which limits their ability to handle the diverse and unconstrained natural language descriptions that arise in real-world scenarios. In contrast, open-vocabulary 3D querying (Kerr et al., 2023) aims to localize and identify objects or regions within a 3D scene based on arbitrary natural language prompts, thereby establishing fine-grained correspondences between linguistic semantics and their spatial locations in three-dimensional space.

Existing open-vocabulary 3D querying methods predominantly adopt a one-shot training and inference paradigm, in which the 3D scene representation is aligned with the semantic knowledge



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.

provided by pretrained vision-language models. Once an initial query is executed, the resulting prediction is typically treated as the final output, without an explicit mechanism for inspecting the intermediate evidence and adapting the subsequent querying procedure. This limitation becomes particularly pronounced in complex scenes, where inaccurate descriptions, occlusion, or ambiguous visual evidence can lead to erroneous localization or semantic matching. When such errors occur, existing approaches generally lack a mechanism to diagnose the source of the failure and use the diagnosis to modify the querying pipeline. Even when rich visual and geometric cues are available to disambiguate the target, these cues are rarely used to determine how the subsequent query should be performed. Consequently, the final result remains strongly dependent on the initial inference procedure.

To overcome the limitations of the one-shot reasoning paradigm, we propose RSI-LGS: Recursive Self-Improvement for Language Gaussian Splatting, an open-vocabulary 3D querying framework with a pipeline-level self-improvement loop. Our key insight is that an intermediate query result should not be treated as the final prediction, but as feedback for adapting the subsequent inference procedure. At each iteration, RSI-LGS first executes the query to obtain candidate results and their associated visual and geometric evidence, and then reviews these intermediate results to diagnose potential localization and semantic matching errors. Based on the diagnosis, the system selects an explicit intervention that modifies the querying pipeline, such as adjusting the view set, clustering configuration, or query formulation, before executing the next iteration. In this way, each iteration converts the outcome of the previous query into an actionable update to the inference state, forming a closed loop of query, review, diagnosis, and refinement.

However, implementing such a pipeline-level self-improvement loop for 3D scenes introduces a practical challenge: a reconstructed scene may contain a large number of observation views, while repeatedly exposing all views to a large multimodal model would quickly exhaust the context budget and incur substantial inference cost. Motivated by recent multimodal large-model systems (Li et al., 2024; An et al., 2025; 2026; Qwen Team, 2026), we therefore propose HALCO (Hierarchical Asymmetric Local-Cloud Orchestration). A local multimodal model handles high-frequency per-view localization and evidence extraction, while only a compact rendering of the current result is passed to the reviewer for once-per-iteration feedback (Tang et al., 2026). By keeping the heavy visual processing local and escalating only the review step, HALCO enables the proposed self-improvement loop to operate within practical context and request budgets.

To comprehensively evaluate the iterative self-improvement capability of open-vocabulary 3D querying, we conduct extensive experiments on established benchmarks, including 3D OVS and LERF-Mask, and compare RSI-LGS against a broad range of representative open-vocabulary 3D querying methods. The results demonstrate that RSI-LGS effectively leverages the review feedback generated during each querying iteration to make targeted adjustments to subsequent queries, leading to continuous improvements in both localization and semantic matching over multiple iterations.

In summary, our main contributions are as follows:

- We propose RSI-LGS, an open-vocabulary 3D querying framework with a pipeline-level self-improvement loop. RSI-LGS reviews intermediate query results, diagnoses localization and semantic matching errors, and adaptively modifies the querying pipeline for subsequent iterations, enabling recursive refinement of the inference process.

- We propose HALCO, an asymmetric local-cloud architecture that makes the self-improvement loop practical for 3D scenes. High-frequency per-view localization and evidence extraction are handled locally, while the more expensive review step is performed by a larger multimodal model, reducing the context and request costs of recursive querying.

- We conduct extensive experiments on the 3D OVS and LERF-Mask benchmarks and compare RSI-LGS with existing open-vocabulary 3D querying methods. The results demonstrate that RSI-LGS achieves competitive performance across multiple evaluation metrics and consistently improves query results through multi-round iterations, validating the effectiveness of the proposed self-improvement loop for open-vocabulary 3D querying.

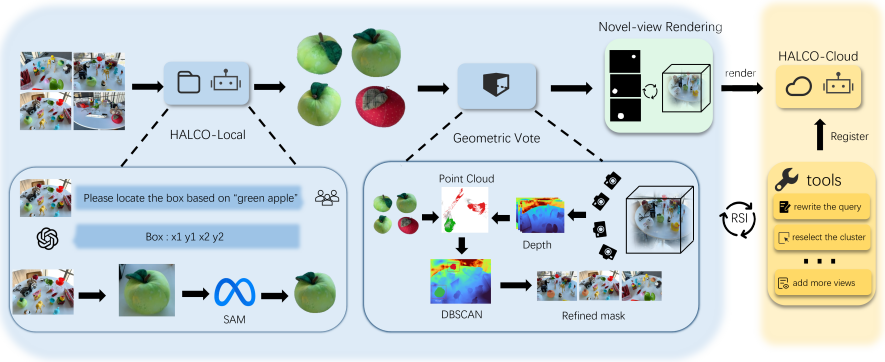



Figure 2: Overview of RSI-LGS, whose two stages are the two tiers of HALCO. Query-conditioned semantic inference on HALCO-Local produces a mask with strong 3D consistency and passes it to the pipeline as feedback; the recursive self-improvement (RSI) loop on HALCO-Cloud then adjusts the whole pipeline according to this feedback and runs it again, closing the loop.

## 2 Related Work

### 2.1 Open-vocabulary 3D scene understanding

Neural Radiance Fields (Mildenhall et al., 2020) demonstrated that photo-realistic novel-view synthesis can be obtained by optimizing a continuous volumetric representation from posed images. The same representation was subsequently used for closed-set semantic labelling and scene understanding (Zhi et al., 2021). Building on it, LERF (Kerr et al., 2023) embedded CLIP (Radford et al., 2021) features into a radiance field and enabled natural-language queries over 3D scenes. This line of work showed the promise of language-conditioned 3D interaction, but NeRF-style rendering and optimization limit scalability and make dense querying expensive.

3D Gaussian Splatting (Kerbl et al., 2023) replaced implicit volumetric rendering with an explicit set of anisotropic Gaussians and a fast differentiable rasterizer, enabling high-quality real-time rendering. Recent work has therefore moved open-vocabulary 3D understanding toward Gaussian representations. LangSplat (Qin et al., 2024) constructs a 3D language field by distilling CLIP-aligned features into language Gaussians, improving both efficiency and object-boundary quality over NeRF-based language fields. Gaussian Grouping (Ye et al., 2024) augments Gaussians with identity encodings supervised by 2D masks, making object-level grouping and editing more practical. OpenGaussian (Wu et al., 2024) further targets point-level open-vocabulary understanding by learning 3D-consistent instance features and associating them with language features. These methods advance feature representation and 3D interaction, but they all rely on one-shot inference results and ignore the self-diagnostic information those results carry about their own errors.

### 2.2 Recursive self-improvement and iterative refinement

Recent work has explored how AI systems can use their own intermediate results or accumulated experience to improve subsequent problem solving, ranging from recursively modifying the improvement procedure itself to iteratively refining task-level outputs. STOP (Zelikman et al., 2023) provides a direct example of recursive self-improvement in code generation, where a language-model-based scaffolding program recursively improves the same scaffolding used to generate and evaluate candidate programs. Agent Workflow Memory (Wang et al., 2024) instead accumulates reusable workflows from previous interactions, allowing experience to influence subsequent task solving without recursively modifying the inference procedure itself.

A separate line of work studies iterative refinement within a single task. ReAct (Yao et al., 2022) interleaves reasoning with tool interaction, Reflexion (Shinn et al., 2023) uses feedback from previous attempts to guide subsequent trials, and Self-Refine (Madaan et al., 2023) critiques and revises intermediate outputs. These methods demonstrate the effectiveness of feedback-driven inference, but primarily refine the reasoning trajectory or output while keeping the underlying improvement

procedure fixed. Moreover, repeated refinement without targeted feedback can lead to redundant or ineffective iterations (Shinn et al., 2023).

RSI-LGS instead performs process-level recursive self-improvement at inference time. Its recursion operates on the 3D querying procedure rather than the underlying model or the current output alone: each query produces visual and geometric evidence, a reviewer diagnoses the failure mode, and the diagnosis determines how the next query is configured. This feedback loop therefore turns intermediate results into explicit modifications of the subsequent inference process, enabling iterative correction of failure modes that may persist under one-shot open-vocabulary 3D querying.

# 3 METHOD

Given a reconstructed 3D Gaussian scene and an open-vocabulary text query, RSI-LGS predicts a query-specific 3D mask that can be rendered from arbitrary viewpoints. The method has two stages, one for each tier of HALCO. The first is a query module built on HALCO-Local: a small multimodal model performs the per-view localization at high frequency, and the resulting masks are consolidated into a 3D-consistent prediction that becomes the feedback of the pipeline. The second is a self-improvement module built on HALCO-Cloud: a large multimodal model reviews the rendered prediction and adjusts the pipeline, which is then re-run, until the prediction passes review or the maximum iteration depth is reached (Figure 2).

## 3.1 PROBLEM DEFINITION

We consider open-vocabulary 3D querying over a pre-reconstructed Gaussian scene. The inputs are the reconstructed 3D Gaussian field $\mathcal{G} = \{g_k\}_{k=1}^{K}$, the multi-view RGB images $\mathcal{I} = \{I_i\}_{i=1}^{N}$ used for reconstruction, and a single open-vocabulary query $q$. The geometry of $\mathcal{G}$ is fixed during querying, and each Gaussian $g_k$ is associated with its spatial and appearance attributes.

The objective is to map the three inputs to a set of relevance scores over the Gaussian field:

$$f_\theta(\mathcal{G}, \mathcal{I}, q) = R = \{r_k\}_{k=1}^{K}, \qquad r_k \in [0, 1],$$

where $R$ denotes the query-conditioned relevance set and $r_k$ measures the semantic relevance between the $k$-th Gaussian $g_k$ and the query $q$. The scores are one-to-one aligned with the Gaussians, so each Gaussian receives exactly one score, and the resulting $R$ localizes the 3D region that corresponds to the query.

## 3.2 QUERY-CONDITIONED SEMANTIC INFERENCE ON HALCO-LOCAL

Given the query $q$, our goal is to obtain reliable 2D semantic masks for the queried object or region while exploiting the underlying 3D Gaussian scene to suppress view-specific errors. To this end, we first obtain query-conditioned 2D localization results from the MLLM and then filter the resulting masks by multi-view 3D geometric consistency.

**View Sampling.** A reconstructed scene is captured from hundreds of posed views, and neighboring views largely observe the same content. The information they carry about the query is therefore highly redundant, and localizing the query on every view would repeat the same computation. We therefore sample a subset $\mathcal{I}_s \subseteq \mathcal{I}$ from the input views at random, and perform all subsequent operations on $\mathcal{I}_s$.

**Semantic Segmentation.** For each selected view $I_i \in \mathcal{I}_s$, we provide the image together with the query $q$ to the MLLM and prompt it to localize the queried object. The MLLM predicts a query-conditioned bounding box $b_i$, which provides a coarse spatial localization of the target in the image. We then feed the predicted bounding box together with the corresponding image $I_i$ into SAM (Kirillov et al., 2023) to obtain a pixel-level segmentation mask:

$$b_i = \mathrm{MLLM}(I_i, q), \qquad M_i = \mathrm{SAM}(I_i, b_i),$$

where $M_i$ denotes the initial semantic mask associated with the query in the $i$-th selected view.

**3D Geometric Voting.** Although the obtained masks provide query-conditioned 2D semantic evidence, they may still contain erroneous or view-specific regions. We therefore introduce a 3D geometric voting process to filter unreliable mask predictions based on the reconstructed Gaussian scene.

Specifically, for each selected view $I_i$, we render a depth map $D_i$ from the pre-reconstructed Gaussian scene $G$. Together with the corresponding camera parameters, the depth map allows us to back-project each image pixel into the 3D scene. For a pixel $p = (x, y)$, its corresponding 3D position is given by

$$\mathbf{X}_i(p) = D_i(p)K^{-1}\begin{bmatrix} x \\ y \\ 1 \end{bmatrix},$$

followed by transformation from the camera coordinate system to the world coordinate system. We consequently obtain a 3D point for every pixel in $I_i$. By applying the initial semantic mask $M_i$, only the 3D points corresponding to the predicted query region are retained:

$$\mathcal{P}_i = \{\mathbf{X}_i(p) \mid M_i(p) = 1\}.$$

The queried object should form a spatially coherent region in the reconstructed 3D scene, while false positives caused by 2D semantic ambiguity tend to be spatially isolated or inconsistent. We therefore apply DBSCAN clustering (Ester et al., 1996) to $\mathcal{P}_i$ and retain only the largest spatial cluster $\mathcal{C}_i^*$. Projecting the retained points back onto the image plane yields a consistency mask $\widetilde{M}_i$. Because clustering keeps only the largest spatial cluster, $\widetilde{M}_i$ can also remove parts of the target, so we treat it as a geometric reference rather than as a replacement for the segmentation mask. The neighborhood radius $\varepsilon$ of DBSCAN sets the spatial scale of this grouping, and it is fixed adaptively for each mask from the distances between in-mask points and their spatial neighbors. A tighter radius keeps only the points that agree most closely in space, which shrinks $\widetilde{M}_i$ and lowers the IoU it reaches against $M_i$, so $\varepsilon$ controls how many views clear the threshold $\tau$.

We then measure how well the original mask $M_i$ agrees with this reference by their intersection-over-union (IoU), $\mathrm{IoU}_i = |M_i \cap \widetilde{M}_i|/|M_i \cup \widetilde{M}_i|$, and accept the view only if $\mathrm{IoU}_i \geq \tau$. The accepted views form the set $\mathcal{I}_a = \{\, i \in \mathcal{I}_s \mid \mathrm{IoU}_i \geq \tau \,\}$, and each accepted view contributes its original mask $M_i$ as the final query-conditioned 2D observation; views below the threshold are discarded. The reconstructed geometry thus decides only which views are trustworthy, while the masks used downstream remain the SAM predictions.

### 3.3 Recursive Self-Improvement Loop on HALCO-Cloud

Although the accepted masks obtained from 2D semantic inference are constrained by the 3D geometry, they may still contain semantic ambiguities or incomplete predictions. To further verify the reliability of the query-conditioned results, we project the accepted 2D observations back into the Gaussian scene and construct a semantic representation that can be rendered from novel viewpoints. This enables us to inspect the predicted 3D semantics from perspectives beyond the original input views and provides additional visual evidence for subsequent error diagnosis.

**Novel-View Rendering.** The query-conditioned prediction is rendered with the differentiable rasterizer of 3D Gaussian Splatting (Kerbl et al., 2023). Instead of a photometric color, each Gaussian carries its relevance score $r_k$, and every pixel of the rendered map is obtained by $\alpha$-blending the $N$ ordered Gaussians that cover it,

$$\hat{M} = \sum_{i \in \mathcal{N}} r_i \alpha_i \prod_{j=1}^{i-1}(1 - \alpha_j),$$

where $r_i$ is the relevance score of the $i$-th Gaussian and $\alpha_i$ is its opacity evaluated through the projection of that Gaussian onto the image plane. All remaining Gaussian attributes are frozen, so the rendered image is a query-conditioned relevance map rather than a photometric one.

Table 1: mIoU and mBIoU on the LERF-Mask benchmark across three scenes. mBIoU dilates both the predicted and the ground-truth masks by a disk of radius $r = \max(1, \mathrm{round}(0.02\sqrt{H^2 + W^2}))$, approximately $2\%$ of the image diagonal, before the IoU is computed (Cheng et al., 2021).

| Scene | RSI-LGS | | Gaussian Grouping | | LangSplat | | OpenGaussian | | ReferSplat | |
|---|---|---|---|---|---|---|---|---|---|---|
| | mIoU | mBIoU | mIoU | mBIoU | mIoU | mBIoU | mIoU | mBIoU | mIoU | mBIoU |
| figurines | **0.8690** | **0.8334** | 0.7311 | 0.7142 | 0.2167 | 0.1935 | 0.1739 | 0.1422 | 0.5445 | 0.5211 |
| ramen | **0.8660** | **0.7595** | 0.6106 | 0.5350 | 0.6049 | 0.5364 | 0.4143 | 0.3271 | 0.7749 | 0.7267 |
| teatime | **0.8653** | **0.7991** | 0.7006 | 0.6556 | 0.2822 | 0.2425 | 0.2757 | 0.1968 | 0.6149 | 0.5641 |
| Overall | **0.8668** | **0.7973** | 0.6808 | 0.6349 | 0.3680 | 0.3241 | 0.2879 | 0.2220 | 0.6448 | 0.6040 |

The relevance field is then fitted to the accepted masks by minimizing an $L_1$ objective over the relevance scores $\{r_k\}$,

$$\mathcal{L} = \sum_{i \in \mathcal{I}_a} \|\hat{M}_i - M_i\|_1,$$

where $\hat{M}_i$ is the relevance map rendered at view $i$ and $M_i$ its accepted mask. We denote the adapted scene by $G^q$, in which only the relevance scores have been fitted while the geometry and the remaining Gaussian parameters stay unchanged. Once fitted, $G^q$ can be rendered from arbitrary novel viewpoints, which lets the review stage inspect the query prediction beyond the views that produced it.

**Review-Driven Self-Improvement.** The reviewer never sees the scene $G^q$ itself. It receives instead the relevance maps rendered from $G^q$ at a set of viewpoints, and these images are the visual feedback for the current round. Together with them we pass a textual description of the pipeline state $S_t$. The state records the number of sampled views $|\mathcal{I}_s|$ and the number of views that survive geometric voting $|\mathcal{I}_a|$, along with other information that describes the pipeline. The same prompt also describes the registered operators that may modify the pipeline. The full state template, the operator list, and the review prompt appear in Appendix A.

We then use a cloud-hosted multimodal large language model $\pi_\theta$ as the reviewer. Given the current state and the registered tools, it invokes one of them to adjust the pipeline, and the pipeline is run again under that adjustment to produce the new state $S_{t+1}$. One round can therefore be written as

$$S_{t+1} = \Phi\big(\pi_\theta(S_t,\, \hat{M}_t,\, \mathcal{T})\big),$$

where $\pi_\theta$ is the reviewer, $\hat{M}_t$ the maps rendered from $G^q$ at round $t$, $\mathcal{T}$ the registered tools, and $\Phi$ the operation that runs the pipeline under the invoked tool. The reviewer picks one tool after reading $\hat{M}_t$ and $S_t$, and the tools it has already tried stay in its context, so each round is decided with the previous attempts in view.

# 4 Experiments

## 4.1 Experimental Setup

**Datasets.** We evaluate RSI-LGS on two public benchmarks for open-vocabulary 3D segmentation, 3D-OVS (Liu et al., 2023) and LERF-Mask (Ye et al., 2024). The two datasets provide complementary evaluation settings for query-conditioned 3D scene understanding.

3D-OVS is a benchmark for weakly supervised 3D open-vocabulary segmentation, providing multi-view RGB images, text descriptions, and camera poses for evaluating text-guided 3D localization.

LERF-Mask is derived from the LERF-Localization dataset (Kerr et al., 2023) and contains three manually annotated scenes: figurines, ramen, and teatime. It provides ground-truth masks for text-guided object localization from novel views.

Table 2: mIoU and mBIoU on the 3D-OVS benchmark across five scenes.

| Scene | RSI-LGS | | Gaussian Grouping | | LangSplat | | OpenGaussian | | ReferSplat | |
|---|---|---|---|---|---|---|---|---|---|---|
| | mIoU | mBIoU | mIoU | mBIoU | mIoU | mBIoU | mIoU | mBIoU | mIoU | mBIoU |
| bed | 0.9392 | 0.8303 | **0.9424** | **0.8416** | 0.7365 | 0.5598 | 0.2822 | 0.2553 | 0.9379 | 0.7876 |
| covered_desk | 0.8581 | **0.7946** | 0.4337 | 0.3913 | 0.7937 | 0.6718 | 0.1850 | 0.1693 | **0.8660** | 0.6684 |
| office_desk | **0.9217** | **0.7531** | 0.7322 | 0.6377 | 0.8091 | 0.6789 | 0.1320 | 0.1213 | 0.4875 | 0.3668 |
| snacks | **0.8969** | **0.8650** | 0.6588 | 0.6084 | 0.8468 | 0.7822 | 0.1666 | 0.1503 | 0.8743 | 0.7876 |
| sofa | **0.9379** | **0.8624** | 0.7174 | 0.6526 | 0.7898 | 0.6969 | 0.2358 | 0.2144 | 0.7644 | 0.6415 |
| Overall | **0.9108** | **0.8211** | 0.6969 | 0.6263 | 0.7952 | 0.6779 | 0.2003 | 0.1821 | 0.7860 | 0.6504 |

**Baselines.** We compare with four representative Gaussian-based methods. LangSplat (Qin et al., 2024) learns a 3D language field by distilling CLIP-aligned language features into Gaussian representations. Gaussian Grouping (Ye et al., 2024) learns object-level identity encodings for Gaussians using SAM-derived 2D masks and spatial regularization. OpenGaussian (Wu et al., 2024) targets point-level open-vocabulary understanding by learning 3D-consistent instance features and associating them with CLIP features. ReferSplat (He et al., 2025) addresses referring segmentation in 3D Gaussian Splatting and conditions the segmentation on positional context, so that spatial relations between the query and the scene can be resolved. We report the mean IoU (mIoU) and the boundary IoU (mBIoU) between the rendered query masks and the ground-truth annotations. All methods are evaluated under the same experimental setup, and the compared methods follow their official implementations.

**Implementation details.** The main hyperparameters are as follows. The IoU consistency rejection threshold is $\tau = 0.5$ and the maximum review depth is $T_{\max} = 10$. For 3D geometric voting, DBSCAN uses $\texttt{min_samples} = 10$ and an adaptive neighborhood radius $\varepsilon$ derived from the distances to neighboring in-mask pixels. View sampling randomly draws 10 frames. Segmentation uses SAM 2.1 (Hiera-Large) (Ravi et al., 2024). Per-view bounding boxes are predicted by LLaVA-OneVision-2-8B-Instruct (An et al., 2026), and the review is performed by DeepSeek V4.1 Flash (DeepSeek-AI, 2026). All experiments run on a single NVIDIA RTX 4090 GPU.

### 4.2 Quantitative Results

Table 1 and Table 2 report the results on LERF-Mask and 3D-OVS, respectively. RSI-LGS attains the best overall result on both benchmarks, demonstrating strong and robust open-vocabulary 3D localization capability. It reaches 0.8668 mIoU on LERF-Mask and 0.9108 on 3D-OVS, gains of $18.6$ and $11.6$ mIoU points over the strongest baseline. The per-scene breakdown is more informative than the averages. On LERF-Mask, RSI-LGS is the best method in every scene and on both metrics, with a per-scene mIoU of 0.8653–0.8690. On 3D-OVS, RSI-LGS operates on sampled views and matches the strongest baseline on each of the two scenes, Gaussian Grouping on bed and ReferSplat on covered_desk, so sampling causes no significant loss of accuracy. On the three harder scenes, office_desk, snacks, and sofa, RSI-LGS still leads by 2.3 to 14.8 mIoU points, which demonstrates the overall stability of the method. We attribute the gains to the iterative query-conditioned reasoning process, which leverages review feedback and 3D geometric consistency to correct erroneous predictions.

### 4.3 Qualitative Analysis

We further provide qualitative comparisons on the LERF-Mask dataset in Fig. 3 and Fig. 4. As shown, RSI-LGS consistently produces more compact and spatially coherent masks, with fewer false-positive regions and more accurate object boundaries across different scenes. In contrast, existing methods often suffer from either incomplete localization or excessive activation beyond the target objects.

Gaussian Grouping relies heavily on 2D SAM-based masks (Kirillov et al., 2023) and Grounded-SAM for text-to-mask matching (Liu et al., 2024). Once the initial text–mask association is incorrect, the resulting error can be propagated into the 3D representation without an explicit mechanism

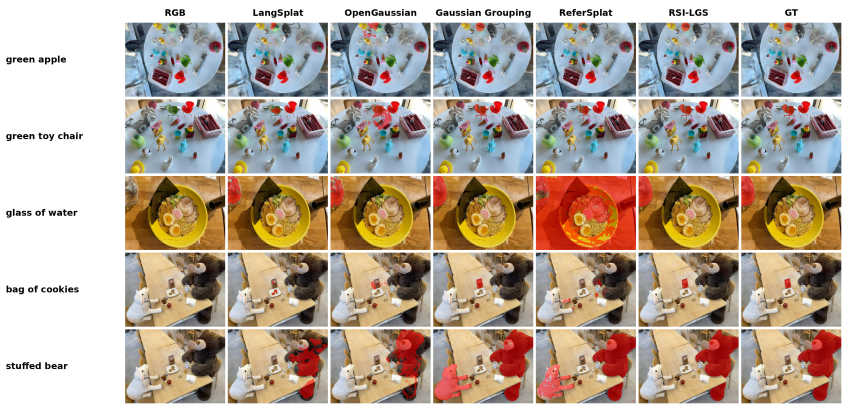



Figure 3: Qualitative comparison on the LERF-Mask benchmark.

for subsequent correction. This leads to incomplete or incorrectly localized regions in several cases. LangSplat learns 3D language features from SAM-derived hierarchical masks and CLIP features. Although this design provides efficient language querying, ambiguities in the 2D semantic features and inaccurate mask boundaries can be propagated into the learned 3D language field, resulting in fragmented or overly expanded predictions. OpenGaussian further relies on SAM-based instance features and an indirect 2D–3D association between 3D points and mask-level CLIP features. Errors in instance grouping or feature association can therefore lead to scattered and spatially inconsistent query responses. ReferSplat attaches a referring feature to every Gaussian and renders the text response directly as the mask. Because nothing constrains the surviving responses to a single spatial component, its masks spread beyond the target, and the errors of the 2D pseudo-masks that supervise the field are frozen into it.

In contrast, RSI-LGS does not treat the initial query as the final prediction. By reviewing the intermediate results, exploiting 3D geometric consistency, and performing visual verification from informative views, RSI-LGS can identify and suppress erroneous semantic evidence before refining the subsequent query. This iterative process produces more spatially coherent and fine-grained masks, which is consistent with the higher mIoU and mBIoU reported in Tables 1 and 2.

### 4.4 Ablation Study

We ablate the two core modules of the pipeline, geometric voting (GV) and recursive self-improvement (RSI), on the figurines scene of LERF-Mask. All configurations are run with identical hyperparameters and query sets, and each number is an average over queries. Two further configurations vary the localizer of the local tier: one removes the local model so that every per-view localization is served by the cloud API, which we denote as w/o HALCO-Local, and one replaces the default localizer with a second local model (Qwen Team, 2026).

**Ablation on Geometric Voting.** Removing geometric voting lowers the mIoU from 0.8690 to 0.8215, a drop of $0.0475$. The loss is highly structured across queries rather than uniform: "green apple" falls from 0.9580 to 0.7389 and "red apple" from 0.9583 to 0.8375. These are exactly the queries whose initial 2D masks contain spatially isolated false-positive regions, which is the failure mode that DBSCAN voting is designed to reject. Without voting, the relevance adaptation is supervised by those stray regions and the rendered relevance field leaks away from the target.

**Ablation on the RSI Loop.** Removing the RSI loop lowers the mIoU from 0.8690 to 0.8210, a drop of $0.0480$ that is comparable to the loss caused by removing geometric voting, but it damages a different set of queries. The clearest case is "rubber duck with red hat", which drops from 0.9094 to 0.6443: its first-round prediction is geometrically coherent and therefore survives voting, yet it is semantically mismatched, and without a review stage nothing detects the error or triggers a second round.

**Complementarity.** Removing both modules yields 0.7706, i.e., $\Delta = -0.0984$, which is close to the sum of the two individual drops; the interaction term is only $-0.0029$. Boundary IoU follows the same pattern, with drops of $0.0324$, $0.0347$, and $0.0872$ at a band ratio of $0.02$, so the two mech-

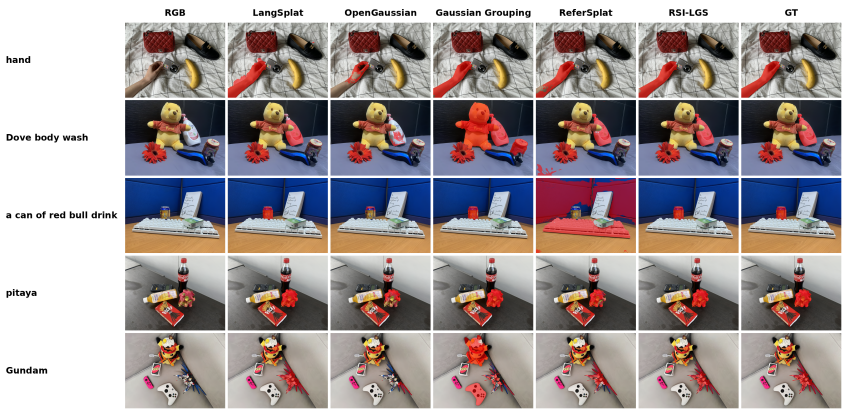



Figure 4: Qualitative comparison on the 3D-OVS benchmark.

anisms improve region coverage and boundary adherence together rather than trading one for the other. Geometric voting and the RSI loop are thus largely additive: voting rejects 2D masks that are inconsistent with the reconstructed 3D geometry, whereas the RSI loop rejects 3D predictions that are geometrically consistent but semantically wrong. The two modules therefore correct different errors across the figurines queries.

**Ablation on HALCO-Local.** In this configuration the localization stage is served by the same API that performs the review, so no local model is used and every per-view localization query is sent to the cloud. The mIoU drops from 0.8690 to 0.7153, a loss of 0.1537 that is larger than the 0.0984 lost when geometric voting and the RSI loop are removed together. Localization errors therefore cannot be recovered by the later voting and review stages: the local tier is what keeps them out of the pipeline in the first place. It also holds down the request budget, since the high-frequency per-view queries never leave the device.

**Ablation on the HALCO-Cloud.** To further investigate the contribution of HALCO-Cloud, we conduct an ablation study by removing the cloud-side reviewer and using the same model as HALCO-Local, LLaVA-OneVision-2-8B-Instruct, for the review process. The resulting performance drops from 0.8690 to 0.7920 mIoU and from 0.8334 to 0.7694 mBIoU, indicating that the local model alone is less effective in handling the reviewer role. This performance degradation suggests that the reviewer requires stronger multimodal reasoning and tool-use capabilities to reliably interpret the intermediate results and invoke the appropriate pipeline operators. The limited model capacity of the local model may constrain its ability to perform these review and tool-selection operations accurately, highlighting the complementary roles of the local and cloud-side models in HALCO.

Table 3: Ablation of geometric voting (GV), recursive self-improvement (RSI), and the two-tier design (HALCO) on the figurines scene of LERF-Mask.

| Configuration | mIoU | | mBIoU | |
|---|---|---|---|---|
| | value | Δ | value | Δ |
| RSI-LGS | **0.8690** | – | **0.8334** | – |
| w/o RSI | 0.8210 | −0.0480 | 0.7987 | −0.0347 |
| w/o GV | 0.8215 | −0.0475 | 0.8010 | −0.0324 |
| w/o GV and RSI | 0.7706 | −0.0984 | 0.7462 | −0.0872 |
| w/o HALCO-Local | 0.7153 | −0.1537 | 0.6929 | −0.1405 |
| w/o HALCO-Cloud | 0.7920 | −0.0770 | 0.7694 | −0.0640 |

## 5 Conclusion

We presented RSI-LGS, which casts open-vocabulary 3D querying as a self-improvement loop: after each round the rendered prediction is reviewed by a cloud-hosted model, which either accepts it or adjusts the pipeline through a registered operator until it does. On LERF-Mask and 3D-OVS, RSI-LGS attains the best overall mIoU and mBIoU, and the ablations show that geometric voting and the RSI loop both contribute. This indicates that reviewing intermediate predictions improves open-vocabulary 3D querying beyond one-shot prediction. We leave a larger operator set and the full benchmarks for future work.

### AI USE STATEMENT

We used generative AI tools as components of the proposed method and for manuscript editing to improve readability. We reviewed all AI-assisted outputs, including those reported in the experiments, and take full responsibility for the final content. No other AI-assisted tasks requiring disclosure were performed.

## REFERENCES

Xiang An, Yin Xie, Kaicheng Yang, Wenkang Zhang, Xiuwei Zhao, Zheng Cheng, Yirui Wang, Songcen Xu, Changrui Chen, Chunsheng Wu, Huajie Tan, Chunyuan Li, Jing Yang, Jie Yu, Xiyao Wang, Bin Qin, Yumeng Wang, Zizhen Yan, Ziyong Feng, Ziwei Liu, Bo Li, and Jiankang Deng. Llava-onevision-1.5: Fully open framework for democratized multimodal training. In *arXiv*, 2025.

Xiang An, Yin Xie, Feilong Tang, Yunyao Yan, Huajie Tan, Didi Zhu, Changrui Chen, Xiuwei Zhao, Bin Qin, Kaicheng Yang, Yifei Shen, Yuanhan Zhang, Kaichen Zhang, Wenkang Zhang, Zheng Cheng, Nansen Zhang, Chunsheng Wu, Chunjiang Ge, Zimin Ran, Dehua Song, Chunyuan Li, Shikun Feng, Ming Hu, Zhangquan Chen, Junbo Niu, Bo Li, Ziyong Feng, Ziwei Liu, Zongyuan Ge, and Jiankang Deng. Llava-onevision-2: Towards next-generation perceptual intelligence. In *arXiv*, 2026.

Bowen Cheng, Ross Girshick, Piotr Dollár, Alexander C. Berg, and Alexander Kirillov. Boundary iou: Improving object-centric image segmentation evaluation, 2021. URL https://arxiv.org/abs/2103.16562.

DeepSeek-AI. Deepseek-v4.1-flash: Pushing the limits of kv cache compression, 2026.

Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, KDD'96, pp. 226–231. AAAI Press, 1996.

Shuting He, Guangquan Jie, Changshuo Wang, Yun Zhou, Shuming Hu, Guanbin Li, and Henghui Ding. Refersplat: Referring segmentation in 3d gaussian splatting. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, volume 267 of *Proceedings of Machine Learning Research*. PMLR / OpenReview.net, 2025. URL https://proceedings.mlr.press/v267/he25h.html.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4), July 2023. URL https://repo-sam.inria.fr/fungraph/3d-gaussian-splatting/.

Justin* Kerr, Chung Min* Kim, Ken Goldberg, Angjoo Kanazawa, and Matthew Tancik. Lerf: Language embedded radiance fields. In *International Conference on Computer Vision (ICCV)*, 2023.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. *arXiv:2304.02643*, 2023.

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. Llava-onevision: Easy visual task transfer. *Transactions on Machine Learning Research*, 2024.

Kunhao Liu, Fangneng Zhan, Jiahui Zhang, Muyu Xu, Yingchen Yu, Abdulmotaleb El Saddik, Christian Theobalt, Eric Xing, and Shijian Lu. Weakly supervised 3d open-vocabulary segmentation. *arXiv preprint arXiv:2305.14093*, 2023.

Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, and Lei Zhang. Grounding dino: Marrying dino with grounded pre-training for open-set object detection, 2024. URL https://arxiv.org/abs/2303.05499.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. Self-refine: Iterative refinement with self-feedback, 2023. URL https://arxiv.org/abs/2303.17651.

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 *ECCV*, 2020.

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*, pp. 20051–20060, 2024.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026. URL https://qwen.ai/blog?id=qwen3.5.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8748–8763. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/radford21a.html.

Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024. URL https://arxiv.org/abs/2408.00714.

Noah Shinn, Federico Cassano, Edward Berman, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning, 2023. URL https://arxiv.org/abs/2303.11366.

Feilong Tang, Xiang An, Yunyao Yan, Yin Xie, Bin Qin, Kaicheng Yang, Yifei Shen, Yuanhan Zhang, Chunyuan Li, Shikun Feng, Changrui Chen, Huajie Tan, Ming Hu, Manyuan Zhang, Bo Li, Ziyong Feng, Ziwei Liu, Zongyuan Ge, and Jiankang Deng. Onevision-encoder: Codec-aligned sparsity as a foundational principle for multimodal intelligence. *arXiv preprint arXiv:2602.08683*, 2026.

Zora Zhiruo Wang, Jiayuan Mao, Daniel Fried, and Graham Neubig. Agent workflow memory, 2024. URL https://arxiv.org/abs/2409.07429.

Yanmin Wu, Jiarui Meng, Haijie Li, Chenming Wu, Yahao Shi, Xinhua Cheng, Chen Zhao, Haocheng Feng, Errui Ding, Jingdong Wang, and Jian Zhang. Opengaussian: Towards point-level 3d gaussian-based open vocabulary understanding. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, pp. 19114–19138, 2024.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*, 2022.

Mingqiao Ye, Martin Danelljan, Fisher Yu, and Lei Ke. Gaussian grouping: Segment and edit anything in 3d scenes. In *ECCV*, 2024.

Eric Zelikman, Eliana Lorch, Lester Mackey, and Adam Tauman Kalai. Self-taught optimizer (stop): Recursively self-improving code generation. *arXiv preprint arXiv:2310.02304*, 2023.

Shuaifeng Zhi, Tristan Laidlow, Stefan Leutenegger, and Andrew J. Davison. In-place scene labelling and understanding with implicit scene representation. 2021.

## A SELF-MODIFICATION OPERATORS AND PROMPTING

We use multimodal large language models (MLLMs) both for per-view bounding-box prediction and for iterative review. For each query, the reviewer receives a textual representation of the current pipeline state, including the number of sampled views, the numbers of candidate and accepted views, the current DBSCAN neighborhood radius $\varepsilon$, and the cluster structure obtained from voting. It also receives the descriptions of six registered operators that expose the available pipeline interventions. Four operators modify the inference procedure: set_eps adjusts the DBSCAN neighborhood radius $\varepsilon$, request_more_views samples additional candidate views, select_cluster selects the spatial cluster to retain, and rewrite_query revises the query formulation. The remaining two operators control termination: finish_review accepts the current prediction, while undo_and_finish reverts the most recent adjustment and accepts the previous state.

At the query stage, the MLLM predicts the referred object's bounding box in a normalized 0–1000 coordinate system, which is then used to initialize a SAM mask proposal for each candidate view. At each review iteration, the reviewer inspects the rendered results and pipeline state, selects an appropriate intervention when necessary, and either continues the loop or terminates with finish_review. The review process is therefore implemented as a sequence of query, review, and pipeline intervention steps, with no parameter updates to the underlying MLLMs.

```
Current query: "<QUERY>"
State:
- input training images   : <N_SAMPLED>
- accepted (voted) frames : <N_ACCEPTED>
- voting eps              : <EPS>
- history                 : <TOOL_CALL_HISTORY>

Figure <K> is the novel-view contact sheet: the left column is
the RGB render and the right column the segmentation, in which
the target region is kept and the rest is filled white.

If Figure <K+1> is attached, it is the cluster contact sheet.
Each tile is one DBSCAN cluster, showing its RGB cropped by
that cluster's mask, with the outside filled white. The tiles
carry NO printed labels; use the tile-to-cluster mapping below
(row-major order, left-to-right, then top-to-bottom) to know
which cluster each tile is:
    tile <T> -> <CLUSTER_ID>  views=<N_VIEWS>  size=<SIZE>
    <one line per tile>
The mapping above lets you refer to each tile's cluster by id
(call select_cluster(<CLUSTER_ID>)). Use it to judge whether
the default cluster really matches the query target; if not,
call select_cluster(<CLUSTER_ID>) with a matching tile that
has more views.

Available tools:
  set_eps(<EPS>)                          - adjust DBSCAN voting
                                            tightness and re-vote
  request_more_views(<QUERY>, <COUNT>)    - acquire more training
                                            views
  rewrite_query(<QUERY>, <NEW_QUERY>)     - rewrite the query
  select_cluster(<CLUSTER_ID>)            - choose a cluster from
                                            Figure <K+1> as the
                                            final supervision
  undo_and_finish(<STATUS>, <REASON>)     - revert the last change
                                            and stop
  finish_review(<STATUS>, <REASON>)       - conclude the review
```

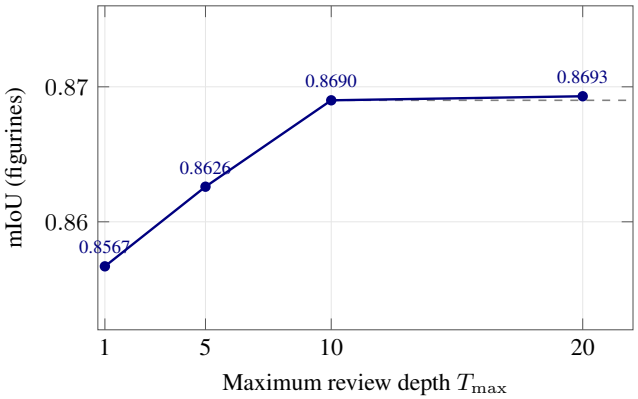



Figure 5: mIoU on the figurines scene of LERF-Mask as a function of the maximum review depth $T_{\max}$. The dashed line marks the $T_{\max} = 10$ level used in our experiments.

# B Iteration Behavior and Runtime Analysis

## B.1 Review Depth Analysis

To keep the loop from running indefinitely, we cap the number of rounds at $T_{\max}$. Figure 5 sweeps this cap on the figurines scene: the mIoU keeps rising from $T_{\max} = 1$ to $T_{\max} = 10$ and then saturates, changing by only $0.0003$ between $T_{\max} = 10$ and $T_{\max} = 20$. We therefore run all experiments with $T_{\max} = 10$, a bound high enough to leave the loop free to continue whenever the review asks for it.

## B.2 Review Tool Usage

Table 4: Review tool usage over the evaluation queries. Every query terminated through an explicit terminating tool call, with no run truncated by the depth limit.

| Tool | Calls (share) |
|---|---|
| `request_more_views` | 54 (42.2%) |
| `finish_review` | 51 (39.8%) |
| `set_eps` | 14 (10.9%) |
| `select_cluster` | 5 (3.9%) |
| `rewrite_query` | 2 (1.6%) |
| `undo_and_finish` | 2 (1.6%) |
| Total | 128 (2.4 per query) |

Table 4 summarizes the six operators invoked by the reviewer across the evaluation queries. The usage is strongly concentrated on a small subset of interventions: request_more_views and set_eps account for more than half of all 128 calls, whereas rewrite_query is invoked only twice. This distribution is consistent with the characteristics of the LERF-Mask and 3D-OVS queries, whose language descriptions generally specify the target object directly. In these cases, the main ambiguity lies in the available visual evidence and the choice of geometric criterion. Both affect how well the target can be separated from surrounding regions. These factors are directly controlled by request_more_views and set_eps, respectively. The relatively small number of select_cluster calls is also consistent with cases in which voting yields a dominant cluster without requiring further disambiguation.

Overall, the observed tool usage indicates that the reviewer does not apply a fixed intervention sequence. Instead, it selects different interventions according to the evidence available in the current query state, with most interventions targeting visual evidence acquisition or geometric refinement.

Table 5: Wall clock in seconds for a single query on the figurines scene, measured on one NVIDIA RTX 4090. The rows follow the pipeline order. HALCO-Cloud covers the review call together with the pipeline work it triggers.

| Stage | Time (s) |
|---|---|
| View sampling | 0.1 |
| HALCO-Local | 23.7 |
| Geometric voting | 9.4 |
| Novel view rendering | 13.7 |
| HALCO-Cloud | 58.4 |
| Total | 105.3 |

### B.3 Per-Stage Runtime

Table 5 reports the wall-clock time of each pipeline stage for a single query on the figurines scene, measured on one NVIDIA RTX 4090. The complete pipeline takes 105.3 seconds. HALCO-Cloud is the most time-consuming stage at 58.4 seconds, followed by HALCO-Local at 23.7 seconds, while geometric voting and novel-view rendering take 9.4 and 13.7 seconds, respectively.

The runtime profile reflects the asymmetric design of HALCO: high-frequency per-view processing is performed locally, whereas the cloud model is used for the review stage and the pipeline operations triggered by its decisions. Thus, the two components have clearly separated computational roles in the recursive querying pipeline.