ACTFUN3D: TASK-CONDITIONED SCENE NARROWING FOR FINE-GRAINED 3D FUNCTIONAL GROUNDING

Anonymous authors
Paper under double-blind review

ABSTRACT

Fine-grained 3D functional grounding aims to identify the physical region that supports a natural-language task, such as the handle to pull, the button to press, or the control to adjust. This is particularly challenging in large 3D scenes, where task-relevant regions occupy only a small fraction of the scene and are easily confused with visually or geometrically similar candidates. We introduce **ActFun3D**, a task-conditioned coarse-to-fine framework that uses task semantics to reduce the effective search space before fine-grained grounding. Given a task, ActFun3D first extracts structured functional semantics describing the target, its context, affordance, and spatial constraints, and uses them to localize a high-recall candidate region over language-aligned 3D features. A trainable local 3D grounder then integrates local geometry, appearance, multi-view visual features, and task semantics to predict the task-relevant functional region at the point level. To further resolve ambiguous predictions, ActFun3D retrieves a compact set of informative posed RGB-D views and projects verified visual evidence back into 3D for multi-view refinement. Experiments on SceneFun3D show that ActFun3D achieves 30.8 AP_{50} in fine-grained 3D functional grounding. Notably, task-conditioned narrowing retains 94.2% of the target while reducing the effective search space to only 15.9% of the scene, demonstrating the effectiveness of narrowing large 3D scenes before fine-grained grounding. **Project page:** https://anonymous.4open.science/w/ActFun3D-Web/

1 INTRODUCTION

Grounding language in 3D becomes substantially more challenging when an instruction refers not to an object category, but to the small physical region that enables an action. An agent asked to "turn on the TV with the remote control," for example, must identify the power button rather than the TV or the remote as a whole. Recent progress in open-vocabulary 3D perception and vision-language grounding has made it possible to query 3D scenes with free-form language and to transfer rich visual semantics from 2D foundation models into 3D representations (Takmaz et al., 2023; Cao et al., 2025). Beyond explicit referring expressions, recent work has also begun to ground higher-level human intentions in 3D scenes (Kang et al., 2025). Functional grounding pushes this problem further: the target is often a fine-grained interactive element whose identity is determined jointly by the task, the surrounding context, and its functional role (Delitzas et al., 2024).

A central difficulty is the mismatch between the scale of the scene and the scale of the functional target. Interactive elements such as handles, buttons, switches, and dials often occupy only a small fraction of a large 3D scene, while repeated objects and visually or geometrically similar structures introduce many plausible candidates. As the scene grows, fine-grained grounding must therefore identify a small task-relevant region from a much larger set of irrelevant or competing geometry. This difficulty is further amplified by the task description itself, which often specifies the intended action without explicitly naming the physical part to be manipulated. A closely related line of work addresses this semantic gap by inferring functional targets from task descriptions and locating them across visual observations (Corsetti et al., 2025), while recent methods further incorporate multimodal reasoning, task-aware visual guidance, and 3D refinement (Wang et al., 2025; He et al., 2026). These advances show that task semantics can provide useful guidance for functional grounding, but leave a more fundamental question: *can task semantics narrow the 3D search space while preserving the functional target, making fine-grained grounding more tractable?*

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.

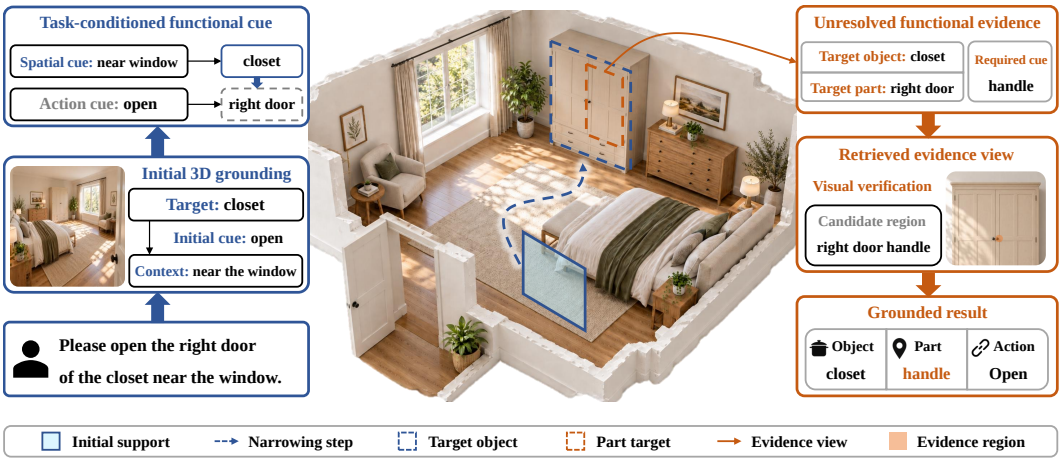


Figure 1: **Task-conditioned scene narrowing for 3D functional grounding.** Fine-grained functional evidence can remain ambiguous in large 3D scenes. Task semantics focus the 3D candidate space and visual evidence search, enabling informative views to verify the functional candidate.

This question suggests a different view of fine-grained functional grounding: task semantics can define *where* grounding should be performed before determining *what* should be grounded. Rather than treating all scene points as equally plausible candidates, the task provides a prior over a task-dependent search region. As illustrated in Figure 1, coarse spatial cues may leave multiple functional interpretations unresolved, whereas task semantics can concentrate the 3D search on a much smaller relevant region and guide subsequent visual evidence toward informative views. We therefore formulate fine-grained 3D functional grounding as a task-conditioned search-space reduction problem, where the coarse stage is designed to suppress task-irrelevant scene content while retaining the functional target. This decomposition leaves the local grounder to resolve fine-grained geometry and appearance within a cleaner and less ambiguous search space.

Based on this formulation, we introduce **ActFun3D**, a task-conditioned coarse-to-fine framework for fine-grained 3D functional grounding. ActFun3D first converts a natural-language task into a structured functional query and matches it with language-aligned 3D point features to identify a high-recall candidate region. A trainable local 3D grounder then combines local geometry, visual features, and task semantics for point-level grounding. The resulting 3D prediction guides the retrieval of a compact set of informative posed RGB-D views, whose verified visual evidence is projected back into 3D for refinement. In this way, semantic narrowing focuses where to search, local grounding resolves the target, and visual evidence corrects remaining ambiguity.

This design directly tests our central hypothesis: scene narrowing should reduce the 3D search space while preserving the functional target, thereby improving fine-grained grounding. We evaluate this trade-off and selective multi-view refinement on SceneFun3D. ActFun3D achieves 30.8 AP_{50}, while structured scene narrowing retains 94.2% of the functional target within only 15.9% of the scene.

Our contributions are summarized as follows:

- We introduce **task-conditioned 3D scene narrowing** for fine-grained functional grounding, using structured task semantics to construct a high-recall search region before local grounding. We evaluate this stage through search-space reduction, target retention, and downstream grounding.
- We develop **ActFun3D**, a coarse-to-fine 3D grounding framework that combines structured functional queries with a trainable LocalFunctionalGrounder. By jointly modeling local geometry, appearance, multi-view visual features, and task semantics within a reduced search space, ActFun3D resolves task-relevant functional regions at the point level.
- We introduce a **selective multi-view evidence refinement** mechanism that retrieves informative and diverse RGB-D views, verifies task relevance, and aggregates visibility-aware positive and negative evidence in 3D. This conservatively refines the initial grounding rather than re-predicting the target from 2D observations.

2 Related Work

2.1 3D Vision-Language Grounding

3D vision-language grounding links natural-language queries to objects and regions in 3D scenes, evolving from referential localization toward open-vocabulary and intention-driven grounding (Liu et al., 2025). Early benchmarks such as ScanRefer and ReferIt3D established language-conditioned object localization in indoor scans (Chen et al., 2020; Achlioptas et al., 2020). Subsequent methods improved grounding through language-guided proposal and point selection (Jain et al., 2022; Luo et al., 2022), spatial and contextual reasoning with finer language–geometry alignment (Chen et al., 2022; Wu et al., 2023b; Yang et al., 2023), and 2D semantic distillation (Bakr et al., 2022). 3D vision-language pre-training and cross-modal consistency further strengthened transferable representations and matching (Zhu et al., 2023; Zhang et al., 2024). Open-vocabulary methods then broadened 3D perception toward semantic and hierarchical region search (Peng et al., 2023; Takmaz et al., 2025). More recently, LIFT-GS scales 3D grounding through 2D vision-language supervision (Cao et al., 2025), while Intent3D extends grounding from explicit references to targets inferred from human intentions (Kang et al., 2025). Functional grounding adds a distinct challenge: the target may be a small interaction region determined jointly by the task and local context. ActFun3D uses task semantics to narrow the 3D search space before resolving this fine-grained target.

2.2 3D Functional and Affordance Understanding

Functional 3D understanding has progressed from modeling object parts and articulated structures to reasoning about where interactions can occur and how elements support actions. Early work established part representations and articulated environments, providing foundations for dense 3D affordance prediction (Deng et al., 2021; Mo et al., 2019; Xiang et al., 2020). Subsequent methods associated local geometry and environmental constraints with actionable regions and interaction primitives, moving toward context-aware affordance understanding (Mo et al., 2021; Geng et al., 2023; Wu et al., 2023a). Language-guided methods further connect semantic instructions with actionable regions, extending affordance prediction toward open-vocabulary and instruction-conditioned interaction (Li et al., 2024; Nguyen et al., 2024b; Zhu et al., 2025; Chu et al., 2025; Yu et al., 2025). Scene-level studies extend functional understanding to realistic environments, modeling interactive elements, functional relations, and task-conditioned functionality (Delitzas et al., 2024; Zhang et al., 2025). Grounding by Remembering (Wang et al., 2026) uses cross-scene and in-scene memory to disambiguate visually similar functional candidates, while UniFunc3D (Lin & Xu, 2026) progressively localizes task-relevant evidence across visual observations. In contrast, ActFun3D explicitly narrows the 3D search space using task semantics before resolving the fine-grained target through local point-level grounding.

2.3 Multi-View 3D Perception and Grounding

Multi-view observations provide complementary appearance and geometric cues for 3D scene understanding. Early RGB-D methods projected image features into 3D or jointly reasoned across 2D and 3D domains, establishing cross-view feature fusion (Dai & Nießner, 2018; Hu et al., 2021). As language grounding emerged in 3D scenes, multi-view modeling was used to reduce viewpoint sensitivity and capture view-dependent spatial relations (Huang et al., 2022; Guo et al., 2023; Huang et al., 2025). Recent work further exploits dense multi-view visual semantics to strengthen language–3D alignment and grounding under partial observations (Zheng et al., 2025). With vision-language foundation models, language-aligned image features, proposals, and masks are increasingly lifted and aggregated in 3D for open-vocabulary retrieval and grounding (Jatavallabhula et al., 2023; Lu et al., 2023; Nguyen et al., 2024a; Huang et al., 2024; Xu et al., 2025). Cross-view consensus further improves the reliability of lifted 2D predictions (Yan et al., 2024). For fine-grained functional grounding, small interaction regions may appear clearly only in limited views, motivating task-relevant view selection before 3D aggregation (Corsetti et al., 2025; He et al., 2026). TASA (He et al., 2026) starts from task-aware 2D affordance cues to guide view selection and subsequent 3D refinement. ActFun3D instead first narrows the language-aligned 3D scene and performs local point-level grounding within the retained region; selected RGB-D views are used only afterward for task-aware verification and conservative refinement.

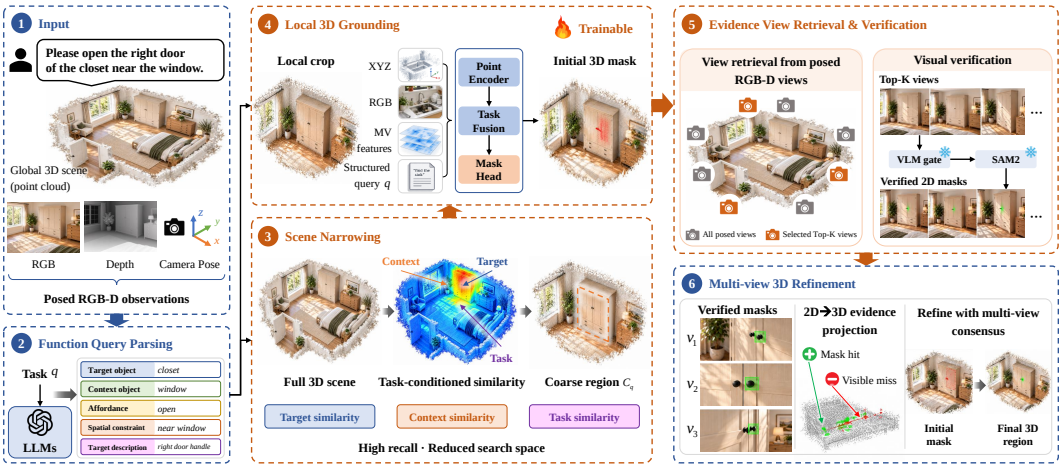


Figure 2: **Overview of ActFun3D.** ActFun3D uses a structured functional query to narrow the full 3D scene to a high-recall candidate region, where a trainable local grounder predicts the fine-grained functional target. The initial prediction guides the retrieval of informative posed RGB-D views, whose verified evidence is projected back into 3D for depth- and visibility-aware refinement.

3 METHOD

3.1 PROBLEM FORMULATION

We consider fine-grained 3D functional grounding following the task setting of SceneFun3D (Delitzas et al., 2024). Given a natural-language task q, a 3D point cloud $\mathcal{P} = \{p_i\}_{i=1}^{N}$, and a fixed set of posed RGB-D observations $\mathcal{V}$, the goal is to identify the physical region that functionally supports the task. We represent this region by a latent target support $\mathcal{Y}_q \subseteq \mathcal{P}$ and predict a point-wise functional score

$$\hat{\mathbf{y}} = F(q, \mathcal{P}, \mathcal{V}), \qquad \hat{\mathbf{y}} \in [0,1]^N, \tag{1}$$

where $\hat{y}_i$ denotes the likelihood that point p_i belongs to $\mathcal{Y}_q$. Unlike an explicitly referred object, $\mathcal{Y}_q$ is determined by the functional requirements of the task and its scene context, and is therefore not known from the task text alone.

Instead of estimating $\mathcal{Y}_q$ over the entire scene directly, we view task semantics as inducing a restricted search domain $\mathcal{P}_q \subseteq \mathcal{P}$. Ideally, this domain should be as compact as possible while retaining the task-relevant support. We formulate this principle as

$$\mathcal{P}_q^* = \arg\min_{\mathcal{P}' \subseteq \mathcal{P}} |\mathcal{P}'| \quad \text{s.t.} \quad \frac{|\mathcal{Y}_q \cap \mathcal{P}'|}{|\mathcal{Y}_q|} \geq \tau, \tag{2}$$

where τ specifies the desired retention of the latent functional support. Since $\mathcal{Y}_q$ is unavailable at inference, Eq. 2 defines an ideal task-conditioned search criterion rather than a directly optimized objective. ActFun3D approximates this criterion by using task semantics to construct a high-recall candidate region, within which fine-grained functional grounding is performed.

As illustrated in Figure 2, this formulation leads to a coarse-to-fine decomposition: task-conditioned scene narrowing first restricts the admissible 3D search domain, local grounding resolves the functional target within this domain, and selected posed RGB-D observations provide complementary evidence for refining the resulting 3D prediction.

3.2 TASK-CONDITIONED SCENE NARROWING

The ideal search domain in Eq. 2 cannot be obtained directly because the functional support $\mathcal{Y}_q$ is unknown at inference time. As illustrated in Figure 2, ActFun3D approximates this domain by extracting structured functional semantics from the task, evaluating their compatibility with the 3D scene, and consolidating high-response points into a task-conditioned search region.

Structured functional query. We parse the task q into a fixed-schema functional query $\mathcal{Q}_q$ organized around the functional target, scene context, and intended interaction. Target semantics are represented by the functional object and target description, contextual semantics by the context object and spatial constraint, and interaction semantics by the affordance and functional relation. The parser does not predict 3D locations or masks; instead, it makes the functional structure of the instruction explicit. For scene narrowing, we use three complementary textual anchors: the target description, the context object, and the original task.

Task-conditioned semantic field. Each scene point p_i is associated with a language-aligned feature $\mathbf{f}_i \in \mathbb{R}^d$ obtained by lifting frozen CLIP representations from the posed RGB-D observations into the 3D scene (Radford et al., 2021). Let $\mathbf{t}_{\mathrm{ctx}}$, $\mathbf{t}_{\mathrm{tgt}}$, and $\mathbf{t}_{\mathrm{task}}$ denote the text embeddings of the context anchor, target anchor, and original task, respectively. After ℓ_2 normalization, the task-conditioned response at point p_i is

$$s_i = \alpha_{\mathrm{ctx}} \mathbf{f}_i^\top \mathbf{t}_{\mathrm{ctx}} + \alpha_{\mathrm{tgt}} \mathbf{f}_i^\top \mathbf{t}_{\mathrm{tgt}} + \alpha_{\mathrm{task}} \mathbf{f}_i^\top \mathbf{t}_{\mathrm{task}}, \tag{3}$$

where α_{ctx}, α_{tgt}, and α_{task} are non-negative weights that sum to one. The target term highlights candidate functional regions, while the context and task terms help suppress semantically plausible but task-irrelevant responses. Together, these cues define a task-conditioned semantic field over the full 3D scene rather than an immediate segmentation.

High-recall spatial consolidation. Semantic responses can remain fragmented across repeated or visually similar structures. We therefore retain the high-response points as semantic seeds,

$$\mathcal{S}_q = \{p_i \in \mathcal{P} \mid s_i \geq \tau_q\}, \tag{4}$$

where τ_q is the score threshold corresponding to a fixed retained fraction of high-response points. We then group $\mathcal{S}_q$ into spatially coherent components using density-based clustering (Ester et al., 1996) and select the component with the strongest semantic support, denoted by $\mathcal{C}_q$.

Because fine-grained functional parts may receive weaker language similarity than their surrounding object, we expand $\mathcal{C}_q$ in 3D and define the task-conditioned search region as

$$\mathcal{P}_q = \left\{ p_i \in \mathcal{P} \;\middle|\; \min_{p_j \in \mathcal{C}_q} \|\mathbf{x}_i - \mathbf{x}_j\|_2 \leq r \right\}, \tag{5}$$

where $\mathbf{x}_i$ is the 3D coordinate of p_i and r is the expansion radius. This expansion favors retention of plausible functional evidence over precise boundary estimation, consistent with the role of $\mathcal{P}_q$ as a search domain rather than a final prediction.

The resulting $\mathcal{P}_q$ removes task-irrelevant scene content while preserving the local geometry and semantic context around plausible targets. The next stage resolves the fine-grained functional region within $\mathcal{P}_q$ using a trainable local 3D grounder. Additional implementation details, including query construction, feature lifting, and spatial consolidation, are provided in Appendix A.1.

3.3 Local 3D Functional Grounding

Given the task-conditioned region $\mathcal{P}_q$ from Eq. 5, we resolve the fine-grained functional target within this reduced domain. Although scene narrowing retains plausible task-relevant evidence, the region still contains nearby geometry and semantically related structures. As shown in Figure 2, we use a trainable local 3D grounder that combines geometry, appearance, lifted visual features, and structured task semantics.

Local point encoding. For each point $p_i \in \mathcal{P}_q$, we construct a local representation from its center-relative coordinate, RGB appearance, and lifted multi-view feature:

$$\mathbf{h}_i^0 = [\mathbf{x}_i - \boldsymbol{\mu}_q; \mathbf{r}_i; \mathbf{f}_i], \qquad \mathbf{z}_i = E_\theta(\mathbf{h}_i^0, \mathcal{P}_q), \tag{6}$$

where $\boldsymbol{\mu}_q$ is the center of $\mathcal{P}_q$, $\mathbf{r}_i$ denotes RGB appearance, $\mathbf{f}_i$ is the lifted visual feature introduced in Section 3.2, and E_θ is the trainable point encoder. Center-relative coordinates preserve local geometry while removing dependence on the absolute scene position.

The encoder spatially orders the local points and partitions them into fixed-size windows. Self-attention is applied within each window with relative positional features, while consecutive blocks alternate regular and shifted partitions. This allows information to propagate across neighboring regions without global attention over the crop, producing the contextual point feature $\mathbf{z}_i$.

Task-conditioned feature fusion. We compose a grounding description from the target description, affordance, context, and spatial constraint in $\mathcal{Q}_q$. A frozen text encoder maps it to a pooled query embedding, which is projected into the point-feature space as $\mathbf{z}_q$. We condition the local point features using feature-wise modulation together with an additive task feature. Let $[\boldsymbol{\gamma}_q, \boldsymbol{\beta}_q]$ denote the modulation parameters predicted from $\mathbf{z}_q$, and let $\mathbf{a}_q$ denote the additive task feature. The conditioned representation is

$$\mathbf{z}_i^c = \mathbf{z}_i \odot (1 + \tanh \boldsymbol{\gamma}_q) + \boldsymbol{\beta}_q + \mathbf{a}_q, \tag{7}$$

where feature-wise modulation adapts the local representation to the task, while $\mathbf{a}_q$ introduces shared task information into the point features.

Point-level prediction and training. To further capture the compatibility between each point and the task, we explicitly model both feature-wise interaction and semantic similarity:

$$\boldsymbol{\phi}_i = [\mathbf{z}_i \odot \mathbf{z}_q; \cos(\mathrm{LN}(\mathbf{z}_i), \mathbf{z}_q)]\,. \tag{8}$$

The original point feature, conditioned feature, task embedding, and point–task interaction are then combined through a residual fusion network, followed by point-wise mask prediction:

$$\widetilde{\mathbf{z}}_i = \mathbf{z}_i + M_\theta\left([\mathbf{z}_i; \mathbf{z}_i^c; \mathbf{z}_q; \boldsymbol{\phi}_i]\right), \qquad \hat{y}_i^{\mathrm{init}} = \sigma(g_\theta(\widetilde{\mathbf{z}}_i))\,, \tag{9}$$

where M_θ is the fusion network, g_θ is the point-wise mask head, and $\hat{y}_i^{\mathrm{init}}$ denotes the initial functional probability of p_i. We optimize the local grounder with a weighted combination of focal and Dice losses to account for the foreground–background imbalance of fine-grained functional regions.

To reduce the gap between training and inference, training samples include both ground-truth-centered crops and regions produced by the same scene-narrowing procedure used at inference time. The pretrained visual and text encoders remain frozen, while the point encoder, query projection, fusion module, and mask head are optimized. The resulting point probabilities $\hat{\mathbf{y}}^{\mathrm{init}} = \{\hat{y}_i^{\mathrm{init}}\}$ form the initial 3D functional prediction and provide the geometric support for the multi-view refinement described next. Additional architectural and training details are provided in Appendix A.2.

3.4 Selective Multi-View Evidence Refinement

Starting from the initial 3D prediction $\hat{\mathbf{y}}^{\mathrm{init}}$ in Eq. 9, we use the posed RGB-D observations to resolve remaining local ambiguities. Rather than aggregating all views, we retrieve a compact set of observations that provide strong and complementary evidence for the predicted functional region. Candidate views are ranked by visibility, projected extent, depth consistency, and viewing geometry, followed by diversity-aware selection to reduce redundant viewpoints. The selected observations are then verified for task relevance, and accepted views are segmented with SAM2 using the projected 3D support as visual guidance, yielding a verified view set $\mathcal{V}_q$ with task-consistent 2D masks.

We project the verified masks back to 3D using the known camera geometry and depth. For point p_i and view v, let $\nu_{iv} \in \{0, 1\}$ indicate reliable visibility, $m_{iv} \in \{0, 1\}$ indicate whether its projection lies inside the verified mask, and $w_{iv} \geq 0$ denote the reliability weight of the observation. We aggregate the multi-view evidence as

$$e_i = \frac{\sum_{v \in \mathcal{V}_q} w_{iv} \nu_{iv} (2m_{iv} - 1)}{\sum_{v \in \mathcal{V}_q} w_{iv} \nu_{iv} + \epsilon}, \tag{10}$$

where mask hits provide positive evidence and visible misses provide negative evidence. Views in which p_i is not reliably visible make no contribution, preventing occlusion or missing observations from being treated as negative evidence.

The aggregated evidence is used to refine $\hat{y}_i^{\mathrm{init}}$ conservatively: consistent visual support preserves the initial prediction, while repeated visible misses suppress unsupported responses. In particular, visual evidence alone does not introduce new positive regions beyond the initial 3D support. This design uses multi-view observations to suppress unsupported responses without replacing the 3D grounder. The refined scores constitute the final functional prediction $\hat{\mathbf{y}}$ defined in Eq. 1. Additional details of view retrieval, verification, evidence projection, and conservative updating are provided in Appendix A.3.

Table 1: Quantitative comparison on SceneFun3D *split0* for fine-grained 3D functional grounding. Best results are shown in **bold**, and second-best results are underlined.

Method	Task-specific Train.	AP_{50} ↑	AP_{25} ↑	mIoU ↑
OpenMask3D (Takmaz et al., 2023)	No	0.2	0.4	0.2
LERF (Kerr et al., 2023)	No	0.0	0.0	0.0
OpenIns3D (Huang et al., 2024)	No	0.0	0.0	0.1
Fun3DU (Corsetti et al., 2025)	No	16.9	33.3	15.2
AffordBot (Wang et al., 2025)	Yes	20.9	24.8	14.4
TASA (He et al., 2026)	Yes	26.9	28.6	19.7
ActFun3D (Ours)	Yes	**30.8**	**41.5**	**23.0**

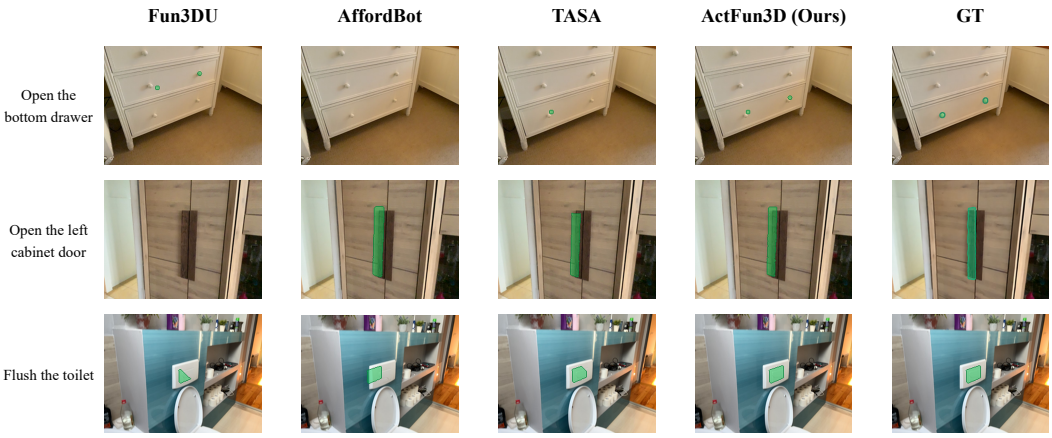


Figure 3: **Qualitative comparison on representative SceneFun3D *split0* examples.** Green overlays denote grounded functional regions. We compare ActFun3D with Fun3DU, AffordBot, and TASA. GT denotes ground truth. Additional comparisons are provided in Appendix C.2.

4 Experiments

4.1 Experimental Setup

Dataset and metrics. We evaluate ActFun3D on SceneFun3D (Delitzas et al., 2024), a benchmark for fine-grained 3D functionality segmentation. We use *split1* for training, with 200 scenes and 2,597 valid tasks, and *split0* for evaluation, with 30 scenes and 445 tasks. One training task with missing or invalid ground-truth annotation is excluded during preprocessing. Following the SceneFun3D Task-2 protocol, we report AP_{50} as the primary metric, with AP_{25} and mIoU measuring localization and point-level overlap. Metrics for scene narrowing are introduced in Sec. 4.3.

Baselines. We compare with open-vocabulary 3D methods, including LERF (Kerr et al., 2023), OpenMask3D (Takmaz et al., 2023), and OpenIns3D (Huang et al., 2024), as well as recent 3D functionality grounding methods, including Fun3DU (Corsetti et al., 2025), AffordBot (Wang et al., 2025), and TASA (He et al., 2026). All baselines are reproduced and evaluated under the same SceneFun3D protocol as ActFun3D. Reproduction details are provided in Appendix B.7.

Implementation details. Scenes are voxelized at $2\,\mathrm{cm}$ resolution, and the local grounder processes up to 16,384 points. The model uses 512-dimensional features, eight encoder layers, 16 attention heads, a window size of 128, and a dropout rate of 0.1. We train the LocalFunctionalGrounder with AdamW using a batch size of 4, a learning rate of 2×10^{-4}, and 1,500 warmup steps. The pretrained language and visual encoders remain frozen, while the LocalFunctionalGrounder is optimized with focal and Dice losses. All experiments are conducted on a single NVIDIA A100 40GB GPU. Further implementation and inference details are provided in Appendix B.

Table 2: Effect of task-conditioned scene narrowing variants on SceneFun3D *split0*. All variants share the same grounder, grounding description, and 16,384-point budget without multi-view refinement; only the candidate region changes. Whole Scene is the unpruned reference.

Scene Narrowing Strategy	SSR (%) ↓	TR (%) ↑	AP_{50} ↑	mIoU ↑
Whole Scene	100.0	100.0	4.2	5.4
Raw Task Query	24.6	80.8	10.7	8.9
Functional Object Query	18.3	89.4	19.8	13.5
Structured Functional Query	**15.9**	**94.2**	**24.1**	**19.1**

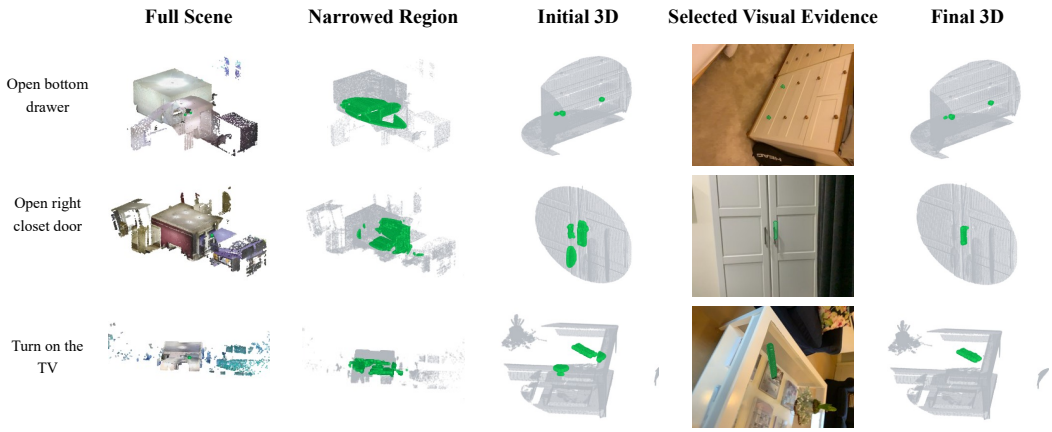


Figure 4: **Task-conditioned scene narrowing and multi-view refinement.** ActFun3D narrows the scene, produces an initial 3D prediction, and refines residual errors using selected visual evidence. Additional examples are provided in Appendix C.3.

4.2 Main Results

Quantitative Results. Table 1 compares reproduced methods on SceneFun3D *split0*. Generic open-vocabulary methods, including OpenMask3D, LERF, and OpenIns3D, perform substantially worse on fine-grained functional grounding. Functionality-oriented methods improve markedly. Fun3DU reaches 33.3 AP_{25} but only 16.9 AP_{50}, indicating better performance at the looser overlap threshold. Among reproduced baselines, TASA leads in AP_{50} and mIoU, and Fun3DU in AP_{25}. ActFun3D achieves 30.8 AP_{50}, 41.5 AP_{25}, and 23.0 mIoU, improving the best reproduced result for each metric by 3.9, 8.2, and 3.3 points, respectively. The AP gains reflect stronger instance-level grounding at both IoU thresholds, while mIoU reflects better point-level overlap.

Qualitative Results. Figure 3 further illustrates the localization differences. For the bottom-drawer task, ActFun3D better matches the sparse interaction regions, while competing methods show incomplete or spatially displaced responses. For the left cabinet door, ActFun3D more closely follows the elongated handle region, whereas competing predictions are incomplete or over-extended. For the toilet-flush task, ActFun3D concentrates on the compact flush control and aligns more closely with the ground truth, while other responses are broader or spatially offset. Overall, the qualitative results are consistent with the quantitative gains, particularly for compact or elongated functional regions that require precise localization. Additional comparisons are provided in Appendix C.2.

4.3 Analysis of Task-Conditioned Scene Narrowing

Narrowing Metrics. We evaluate scene narrowing using two metrics: *Search-Space Ratio* (SSR), defined as $\mathrm{SSR} = |\mathcal{P}_q|/|\mathcal{P}|$, and *Target Retention* (TR), defined as $\mathrm{TR} = |\mathcal{Y}_q \cap \mathcal{P}_q|/|\mathcal{Y}_q|$, where $\mathcal{P}$ is the full scene, $\mathcal{P}_q$ is the retained candidate region, and $\mathcal{Y}_q$ is the ground-truth functional region. Lower SSR indicates a smaller search space, while higher TR indicates better target preservation. Ground truth is used only for post-hoc TR evaluation and is never involved in candidate generation.

Table 3: Component ablation and runtime of multi-view refinement on SceneFun3D *split0*. All variants start from the same initial 3D prediction and use the same geometry-guided view retrieval.

Configuration	SAM2	Verifier	Pos./Neg.	AP_{50} ↑	AP_{25} ↑	mIoU ↑	Time (s)
Initial 3D	✗	–	✗	24.1	26.2	19.1	16.3
Visual Mask Refinement	✓	None	✗	25.7	29.0	20.7	58.5
Geometry Verification	✓	Geometry	✗	26.2	30.6	20.9	49.7
VLM Verification	✓	VLM	✗	29.6	38.3	22.4	112.0
Full ActFun3D	✓	VLM	✓	**30.8**	**41.5**	**23.0**	126.6

Quantitative Analysis. Table 2 shows that structured task semantics provide a better balance between search-space reduction and target preservation. The Raw Task Query retains 24.6% of the scene and 80.8% of the target, while the Functional Object Query reduces SSR to 18.3% and increases TR to 89.4%. The Structured Functional Query further reduces SSR to 15.9% while increasing TR to 94.2%, AP_{50} from 19.8 to 24.1, and mIoU from 13.5 to 19.1. Thus, structured semantics preserve more target evidence despite retaining a smaller region. Relative to Whole Scene under the same grounder and point budget, AP_{50} increases from 4.2 to 24.1 and mIoU from 5.4 to 19.1 while the search space is reduced by 84.1%. These results isolate the benefit of task-conditioned candidate-region construction before fine-grained local grounding.

Qualitative Analysis. Figure 4 further illustrates this behavior. Across different tasks, scene narrowing removes large amounts of irrelevant geometry while retaining small or thin interaction regions, providing a more focused input to the local grounder. The initial 3D predictions may still contain broad or disconnected responses, which are subsequently suppressed using verified multi-view evidence. This effect is particularly clear for thin handles and compact control regions, where the final predictions become more spatially concentrated while preserving the task-relevant target.

4.4 Ablation Study

Table 3 examines the multi-view refinement components. SAM2-based visual masks improve AP_{50} from 24.1 to 25.7, while geometry and VLM verification raise it to 26.2 and 29.6, respectively. VLM verification outperforms the geometry-based verifier by 3.4 points in AP_{50} and 7.7 points in AP_{25}, showing the benefit of task-aware visual filtering. Replacing simple projected-mask gating with visibility-aware signed evidence improves the VLM-verified result by 1.2 AP_{50}, 3.2 AP_{25}, and 0.6 mIoU, yielding $30.8/41.5/23.0$. Diagnostics in Table 6 support the conservative design: precision increases from 0.303 to 0.390 and F1 from 0.354 to 0.403, while recall changes only slightly from 0.425 to 0.417, indicating that refinement primarily suppresses unsupported responses. Runtime shows an accuracy–efficiency trade-off: the initial prediction takes 16.3 s/task, compared with 58.5 s/task for mask-only refinement, 49.7 s/task for geometry verification, 112.0 s/task for VLM verification, and 126.6 s/task for the full model. Geometry verification is faster than mask-only refinement because it filters views before SAM2 segmentation.

5 Conclusion

We presented **ActFun3D**, a task-conditioned coarse-to-fine framework for fine-grained 3D functional grounding. By reducing the scene to a high-recall task-relevant region before local grounding, ActFun3D focuses prediction on compact functional targets and further resolves ambiguity through selective multi-view visual evidence. ActFun3D achieves 30.8 AP_{50}, improving the strongest reproduced baseline by 3.9 points. Notably, structured scene narrowing preserves 94.2% of target points within only 15.9% of the scene. These results support task-conditioned search-space reduction as an effective principle for accurate fine-grained functional grounding in complex 3D scenes.

Limitations and Future Work. ActFun3D relies on posed RGB-D observations, while narrowing errors may discard weak or ambiguous cues. VLM verification adds inference cost. Future work will explore efficient task-aware verification, adaptive view acquisition, and robust narrowing under incomplete observations. Additional failure cases are discussed in Appendix D.

AI USE STATEMENT

Generative AI tools were used in a supporting role during this work. After the authors established the research direction and identified related literature, GPT-5.6 Sol was used to assist in surveying relevant work; all cited references were verified against their sources. It was also used to provide feedback on method and experimental design and for language polishing. All methodological decisions, experimental settings, and results were determined and verified by the authors.

Generative AI models are components of the method. As described in Appendix A.1, a local Qwen3-8B model is used for Function Query Parsing. In the multi-view refinement stage (Appendix A.3), `qwen3.8-flash`, accessed through an API, is used for view verification. The authors reviewed all AI-assisted content and take responsibility for the content and claims of this paper.

ETHICS STATEMENT

This work uses the publicly available SceneFun3D benchmark and does not collect new human-subject data. We cite the official SceneFun3D benchmark and associated resources and follow the dataset's stated usage requirements.

REPRODUCIBILITY STATEMENT

We provide detailed descriptions of the ActFun3D pipeline and experimental procedure to facilitate reproducibility. Additional method details are provided in Appendices A.1–A.3, with dataset processing and key implementation settings described in Appendix B, and evaluation protocols in Appendix B.4. These sections describe the complete experimental pipeline and specify the key settings and parameter choices used in our experiments. Our anonymized **project page** provides a link to an anonymized code repository, through which reviewers can access selected implementation code for key components of the method, together with a `README.md` containing usage instructions for paper review and reproducibility.

REFERENCES

Panos Achlioptas, Ahmed Abdelreheem, Fei Xia, Mohamed Elhoseiny, and Leonidas J. Guibas. ReferIt3D: Neural listeners for fine-grained 3D object identification in real-world scenes. In *Computer Vision – ECCV 2020*, pp. 422–440, 2020.

Eslam Bakr, Yasmeen Alsaedy, and Mohamed Elhoseiny. Look around and refer: 2D synthetic semantics knowledge distillation for 3D visual grounding. In *Advances in Neural Information Processing Systems*, volume 35, pp. 37146–37158, 2022.

Ang Cao, Sergio Arnaud, Oleksandr Maksymets, Jianing Yang, Ayush Jain, Ada Martin, Vincent-Pierre Berges, Paul Mcvay, Ruslan Partsey, Aravind Rajeswaran, Franziska Meier, Justin Johnson, Jeong Joon Park, and Alexander Sax. From thousands to billions: 3D visual language grounding via render-supervised distillation from 2D VLMs. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 6505–6521. PMLR, 2025.

Dave Zhenyu Chen, Angel X. Chang, and Matthias Nießner. ScanRefer: 3D object localization in RGB-D scans using natural language. In *Computer Vision – ECCV 2020*, pp. 202–221, 2020.

Shizhe Chen, Pierre-Louis Guhur, Makarand Tapaswi, Cordelia Schmid, and Ivan Laptev. Language conditioned spatial relation reasoning for 3D object grounding. In *Advances in Neural Information Processing Systems*, volume 35, pp. 20522–20535, 2022.

Hengshuo Chu, Xiang Deng, Qi Lv, Xiaoyang Chen, Yinchuan Li, Jianye Hao, and Liqiang Nie. 3D-AffordanceLLM: Harnessing large language models for open-vocabulary affordance detection in 3D worlds. In *International Conference on Learning Representations (ICLR)*, 2025.

Jaime Corsetti, Francesco Giuliari, Alice Fasoli, Davide Boscaini, and Fabio Poiesi. Functionality understanding and segmentation in 3D scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 24550–24559, 2025.

Angela Dai and Matthias Nießner. 3DMV: Joint 3D-multi-view prediction for 3D semantic scene segmentation. In *Computer Vision – ECCV 2018*, pp. 452–468, 2018.

Alexandros Delitzas, Ayça Takmaz, Federico Tombari, Robert Sumner, Marc Pollefeys, and Francis Engelmann. SceneFun3D: Fine-grained functionality and affordance understanding in 3D scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14531–14542, 2024.

Shengheng Deng, Xun Xu, Chaozheng Wu, Ke Chen, and Kui Jia. 3D AffordanceNet: A benchmark for visual object affordance understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1778–1787, 2021.

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*, pp. 226–231, 1996.

Haoran Geng, Helin Xu, Chengyang Zhao, Chao Xu, Li Yi, Siyuan Huang, and He Wang. GAPartNet: Cross-category domain-generalizable object perception and manipulation via generalizable and actionable parts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7081–7091, 2023.

Zoey Guo, Yiwen Tang, Ray Zhang, Dong Wang, Zhigang Wang, Bin Zhao, and Xuelong Li. ViewRefer: Grasp the multi-view knowledge for 3D visual grounding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 15372–15383, 2023.

Lian He, Meng Liu, Qilang Ye, Yu Zhou, Xiang Deng, and Gangyi Ding. Task-aware 3D affordance segmentation via 2D guidance and geometric refinement. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(6):4654–4662, 2026.

Wenbo Hu, Hengshuang Zhao, Li Jiang, Jiaya Jia, and Tien-Tsin Wong. Bidirectional projection network for cross dimension scene understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14373–14382, 2021.

Ronggang Huang, Haoxin Yang, Yan Cai, Xuemiao Xu, Huaidong Zhang, and Shengfeng He. ViewSRD: 3D visual grounding via structured multi-view decomposition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9726–9736, 2025.

Shijia Huang, Yilun Chen, Jiaya Jia, and Liwei Wang. Multi-view transformer for 3D visual grounding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15524–15533, 2022.

Zhening Huang, Xiaoyang Wu, Xi Chen, Hengshuang Zhao, Lei Zhu, and Joan Lasenby. OpenIns3D: Snap and lookup for 3D open-vocabulary instance segmentation. In *Computer Vision – ECCV 2024*, pp. 169–185, 2024.

Ayush Jain, Nikolaos Gkanatsios, Ishita Mediratta, and Katerina Fragkiadaki. Bottom up top down detection transformers for language grounding in images and point clouds. In *Computer Vision – ECCV 2022*, pp. 417–433, 2022.

Krishna Murthy Jatavallabhula, Alihusein Kuwajerwala, Qiao Gu, Mohd Omama, Ganesh Iyer, Soroush Saryazdi, Tao Chen, Alaa Maalouf, Shuang Li, Nikhil Varma Keetha, Ayush Tewari, Joshua Tenenbaum, Celso de Melo, Madhava Krishna, Liam Paull, Florian Shkurti, and Antonio Torralba. ConceptFusion: Open-set multimodal 3D mapping. In *Proceedings of Robotics: Science and Systems*, 2023.

Weitai Kang, Mengxue Qu, Jyoti Kini, Yunchao Wei, Mubarak Shah, and Yan Yan. Intent3D: 3D object detection in RGB-D scans based on human intention. In *International Conference on Learning Representations (ICLR)*, 2025.

Justin Kerr, Chung Min Kim, Ken Goldberg, Angjoo Kanazawa, and Matthew Tancik. Lerf: Language embedded radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 19729–19739, 2023.

Yicong Li, Na Zhao, Junbin Xiao, Chun Feng, Xiang Wang, and Tat-Seng Chua. LASO: Language-guided affordance segmentation on 3D object. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14251–14260, 2024.

Jiaying Lin and Dan Xu. UniFunc3D: Unified active spatial-temporal grounding for 3D functionality segmentation. *arXiv preprint arXiv:2603.23478*, 2026.

Daizong Liu, Yang Liu, Wencan Huang, and Wei Hu. A survey on text-guided 3-D visual grounding: Elements, recent advances, and future directions. *IEEE Transactions on Neural Networks and Learning Systems*, 36(10):17717–17737, 2025.

Shiyang Lu, Haonan Chang, Eric Pu Jing, Abdeslam Boularias, and Kostas Bekris. OVIR-3D: Open-vocabulary 3D instance retrieval without training on 3D data. In *Proceedings of The 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pp. 1610–1620. PMLR, 2023.

Junyu Luo, Jiahui Fu, Xianghao Kong, Chen Gao, Haibing Ren, Hao Shen, Huaxia Xia, and Si Liu. 3D-SPS: Single-stage 3D visual grounding via referred point progressive selection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16454–16463, 2022.

Kaichun Mo, Shilin Zhu, Angel X. Chang, Li Yi, Subarna Tripathi, Leonidas J. Guibas, and Hao Su. PartNet: A large-scale benchmark for fine-grained and hierarchical part-level 3D object understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 909–918, 2019.

Kaichun Mo, Leonidas J. Guibas, Mustafa Mukadam, Abhinav Gupta, and Shubham Tulsiani. Where2Act: From pixels to actions for articulated 3D objects. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 6813–6823, 2021.

Phuc Nguyen, Tuan Duc Ngo, Evangelos Kalogerakis, Chuang Gan, Anh Tran, Cuong Pham, and Khoi Nguyen. Open3DIS: Open-vocabulary 3D instance segmentation with 2D mask guidance. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4018–4028, 2024a.

Toan Nguyen, Minh Nhat Vu, Baoru Huang, Tuan Van Vo, Vy Truong, Ngan Le, Thieu Vo, Bac Le, and Anh Nguyen. Language-conditioned affordance-pose detection in 3D point clouds. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3071–3078, 2024b.

Songyou Peng, Kyle Genova, Chiyu Jiang, Andrea Tagliasacchi, Marc Pollefeys, and Thomas Funkhouser. OpenScene: 3D scene understanding with open vocabularies. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 815–824, 2023.

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 *Proceedings of the 38th International Conference on Machine Learning*, volume 139, pp. 8748–8763, 2021.

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. In *International Conference on Learning Representations*, 2025.

Ayça Takmaz, Elisabetta Fedele, Robert Sumner, Marc Pollefeys, Federico Tombari, and Francis Engelmann. OpenMask3D: Open-vocabulary 3D instance segmentation. In *Advances in Neural Information Processing Systems*, volume 36, pp. 68367–68390, 2023.

Ayça Takmaz, Alexandros Delitzas, Robert W. Sumner, Francis Engelmann, Johanna Wald, and Federico Tombari. Search3D: Hierarchical open-vocabulary 3D segmentation. *IEEE Robotics and Automation Letters*, 10(3):2558–2565, 2025.

Qirui Wang, Jingyi He, Yining Pan, Xulei Yang, and Shijie Li. Grounding by remembering: Cross-scene and in-scene memory for 3d functional affordances. *arXiv preprint arXiv:2605.11616*, 2026.

Xinyi Wang, Xun Yang, Yanlong Xu, Yuchen Wu, Zhen Li, and Na Zhao. AffordBot: 3D fine-grained embodied reasoning via multimodal large language models. In *Advances in Neural Information Processing Systems*, volume 38, pp. 140367–140387, 2025.

Ruihai Wu, Kai Cheng, Yan Shen, Chuanruo Ning, Guanqi Zhan, and Hao Dong. Learning environment-aware affordance for 3D articulated object manipulation under occlusions. In *Advances in Neural Information Processing Systems*, volume 36, pp. 60966–60983, 2023a.

Yanmin Wu, Xinhua Cheng, Renrui Zhang, Zesen Cheng, and Jian Zhang. EDA: Explicit text-decoupling and dense alignment for 3D visual grounding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19231–19242, 2023b.

Fanbo Xiang, Yuzhe Qin, Kaichun Mo, Yikuan Xia, Hao Zhu, Fangchen Liu, Minghua Liu, Hanxiao Jiang, Yifu Yuan, He Wang, Li Yi, Angel X. Chang, Leonidas J. Guibas, and Hao Su. SAPIEN: A simulated part-based interactive environment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11097–11107, 2020.

Runsen Xu, Zhiwei Huang, Tai Wang, Yilun Chen, Jiangmiao Pang, and Dahua Lin. VLM-Grounder: A VLM agent for zero-shot 3D visual grounding. In *Proceedings of The 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 3961–3985. PMLR, 2025.

Mi Yan, Jiazhao Zhang, Yan Zhu, and He Wang. MaskClustering: View consensus based mask graph clustering for open-vocabulary 3D instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 28274–28284, 2024.

An Yang et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Li Yang, Chunfeng Yuan, Ziqi Zhang, Zhongang Qi, Yan Xu, Wei Liu, Ying Shan, Bing Li, Weiping Yang, Peng Li, Yan Wang, and Weiming Hu. Exploiting contextual objects and relations for 3D visual grounding. In *Advances in Neural Information Processing Systems*, volume 36, pp. 49542–49554, 2023.

Chunlin Yu, Hanqing Wang, Ye Shi, Haoyang Luo, Sibei Yang, Jingyi Yu, and Jingya Wang. SeqAfford: Sequential 3D affordance reasoning via multimodal large language model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1691–1701, 2025.

Chenyangguang Zhang, Alexandros Delitzas, Fangjinhua Wang, Ruida Zhang, Xiangyang Ji, Marc Pollefeys, and Francis Engelmann. Open-vocabulary functional 3D scene graphs for real-world indoor spaces. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19401–19413, 2025.

Yachao Zhang, Runze Hu, Ronghui Li, Yanyun Qu, Yuan Xie, and Xiu Li. Cross-modal match for language conditioned 3D object grounding. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(7):7359–7367, 2024.

Henry Zheng, Hao Shi, Qihang Peng, Yong Xien Chng, Rui Huang, Yepeng Weng, Zhongchao Shi, and Gao Huang. DenseGrounding: Improving dense language-vision semantics for ego-centric 3D visual grounding. In *International Conference on Learning Representations (ICLR)*, 2025.

He Zhu, Quyu Kong, Kechun Xu, Xunlong Xia, Bing Deng, Jieping Ye, Rong Xiong, and Yue Wang. Grounding 3D object affordance with language instructions, visual observations and interactions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 17337–17346, 2025.

Ziyu Zhu, Xiaojian Ma, Yixin Chen, Zhidong Deng, Siyuan Huang, and Qing Li. 3D-VisTA: Pre-trained transformer for 3D vision and text alignment. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 2911–2921, 2023.

APPENDIX

A ADDITIONAL METHOD DETAILS

This section provides additional methodological and implementation details for Secs. 3.2–3.4, covering scene narrowing, local grounding, and multi-view refinement. Numerical settings are summarized in Appendix B.

A.1 TASK-CONDITIONED SCENE NARROWING

Functional Query Parsing. Given only the task instruction, we use a locally deployed Qwen3-8B model to instantiate the structured query defined in Sec. 3.2. The parser extracts the target description, functional context, affordance, and spatial constraints, while distinguishing the object whose state changes from the physical part that must be manipulated. It also accounts for tools or remote controls when they mediate the interaction. The target description is constrained to the smallest relevant physical region, while spatial relations are retained to disambiguate repeated objects. Generation is deterministic; parsed outputs are validated as structured JSON and cached so that each task uses a fixed query throughout the pipeline.

Multi-View Feature Lifting. The language-aligned feature $\mathbf{f}_i$ used in Eq. 3 is obtained by lifting dense CLIP features from the posed RGB-D observations. A point contributes to a view only when its projection is valid and depth-consistent with the observation. Let $\mathcal{V}_i$ denote the valid views of point $\mathbf{x}_i$ and $\phi_v(\pi_v(\mathbf{x}_i))$ its image feature in view v. We compute

$$\mathbf{f}_i = \mathrm{Norm}\left(\frac{\sum_{v\in\mathcal{V}_i} w_{iv}\phi_v(\pi_v(\mathbf{x}_i))}{\sum_{v\in\mathcal{V}_i} w_{iv} + \epsilon}\right), \tag{11}$$

where w_{iv} reflects depth consistency. The lifted features are precomputed for each scene and reused throughout the pipeline.

Robust Spatial Consolidation. The density-based clustering in Sec. 3.2 may produce multiple candidate components. We rank each component C using

$$S(C) = \frac{1}{|C|}\sum_{i\in C} s_i + \lambda_{\text{size}}\log(1+|C|), \tag{12}$$

which favors strong semantic responses while giving a small preference to spatially supported components. The highest-scoring component is expanded according to Eq. 5. If clustering produces no valid component, the semantic seeds are retained directly. For unusually small or large candidate regions, predefined expansion and score-based fallback rules are applied to maintain a usable local search region. All retained-fraction thresholds, clustering parameters, and fallback rules are fixed across scenes and tasks and do not use ground-truth functional labels; the core numerical settings are summarized in Table 5.

A.2 LOCAL 3D FUNCTIONAL GROUNDING

Task-Conditioned Fusion. The pooled task embedding is projected into the point-feature space and used to condition local point features through feature-wise modulation and an additive task feature. We further model explicit point–task interactions using element-wise feature products and cosine compatibility. The original point feature, conditioned feature, task embedding, and interaction features are then combined through residual fusion for point-wise functional prediction. This formulation injects task semantics into each local point representation without updating the pretrained visual or text encoders.

Training Objective. Let $y_i \in \{0,1\}$ denote the functional label of point p_i and $\hat{y}_i^{\text{init}}$ the prediction from Eq. 9. The local grounder is optimized with

$$\mathcal{L}_{\text{ground}} = \lambda_{\text{f}}\mathcal{L}_{\text{focal}} + \lambda_{\text{d}}\mathcal{L}_{\text{dice}}, \tag{13}$$

where focal loss addresses the foreground–background imbalance and Dice loss encourages overlap with sparse functional targets. Only the point encoder, query projection, fusion module, and mask head are optimized, while the pretrained visual and text encoders remain frozen. To reduce the train–inference gap, training uses both ground-truth-centered regions and regions produced by the same scene-narrowing procedure used at inference time; detailed training settings are provided in Appendix B.2.

Fixed-Budget Inference. The local grounder operates under a fixed point budget. When the narrowed region $\mathcal{P}_q$ exceeds this budget, points are subsampled while preserving high-response regions and spatial coverage. Predictions are propagated to the remaining points in $\mathcal{P}_q$ by nearest-neighbor assignment in 3D, while points outside $\mathcal{P}_q$ receive zero initial probability. Finally, predictions on the voxelized representation are mapped back to the original Faro points for multi-view refinement and evaluation. Detailed inference settings are provided in Appendix B.3.

A.3 Multi-View Evidence Refinement

Geometry-Guided View Retrieval. We first obtain the initial 3D support $\mathcal{T}$ from $\hat{\mathbf{y}}^{\text{init}}$. For each posed RGB-D view v, we measure how well it observes $\mathcal{T}$ using

$$R_v = \alpha\rho_v^{\text{vis}} + \beta\rho_v^{\text{area}} + \gamma\rho_v^{\text{depth}} + \delta\rho_v^{\text{front}} + \eta\exp(-d_v/\sigma_d), \tag{14}$$

where the terms measure visible support, projected extent, depth consistency, viewing frontality, and camera–target distance, respectively. Core inference settings are fixed across tasks and summarized in Table 5. Views are ranked by R_v, followed by diversity-aware selection to reduce redundant viewpoints.

View Verification and Mask Extraction. In the full model, the retrieved views are verified using `qwen3.8-flash`. The verifier receives the task semantics together with the image and projected 3D support, and rejects views that do not provide visual evidence for the functional region. Accepted views are passed to SAM2, using the projected support as the visual prompt, to obtain the 2D masks used in Eq. 10. Rejected views do not contribute to evidence aggregation. The binary decision b_v is determined by the verifier's `relevant` field; the returned confidence score is not used as an additional acceptance threshold.

Refinement Ablation Variants. All variants in Table 3 share the same initial 3D prediction and geometry-guided view retrieval. Visual Mask Refinement applies SAM2 to the retrieved views without verification. Geometry Verification filters the retrieved views using the geometry-based verifier, while VLM Verification uses the task-aware VLM verifier. For these three variants, the accepted 2D masks are projected back to 3D to form a binary positive-support map,

$$s_i^+ = \max_{v\in\mathcal{V}_q} m_{iv},$$

and the initial prediction is refined by simple mask gating,

$$\hat{y}_i^{\text{mask}} = \hat{y}_i^{\text{init}} s_i^+.$$

This baseline does not explicitly distinguish visible mask misses from points lacking reliable observations. Full ActFun3D instead models this distinction through the visibility-aware signed evidence in Eq. 10: mask hits provide positive evidence, reliably visible misses provide negative evidence, and unreliable or unobserved points contribute no evidence.

Conservative 3D Refinement. Let $e_i \in [-1, 1]$ denote the signed evidence from Eq. 10, and normalize it as $u_i = (e_i + 1)/2$. For points with sufficient reliable observations, we update

$$\hat{y}_i = \begin{cases} \min\big(\hat{y}_i^{\text{init}}, \lambda_{\text{3D}}\hat{y}_i^{\text{init}} + \lambda_{\text{vis}}u_i\big), & n_i \geq n_{\min}, \\ \hat{y}_i^{\text{init}}, & \text{otherwise}, \end{cases} \tag{15}$$

where n_i is the number of reliable observations of point p_i, $n_{\min} = 2$ is the minimum number of observations required for refinement, and $\lambda_{\text{3D}}, \lambda_{\text{vis}} \geq 0$ with $\lambda_{\text{3D}} + \lambda_{\text{vis}} = 1$. The outer minimum makes the refinement explicitly conservative: consistent positive evidence preserves an existing 3D response, whereas repeated visible misses can suppress unsupported predictions. Visual evidence alone cannot increase the initial score or introduce a new positive region. Points without sufficient evidence remain unchanged. Numerical settings are provided in Appendix B.3.

Algorithm 1: Selective Multi-View Evidence Refinement

Input: Task q, structured query $\mathcal{Q}_q$, initial scores $\hat{\mathbf{y}}^{\text{init}}$, scene points $\mathcal{P}$, and posed RGB-D views $\mathcal{V}$
Output: Refined functional scores $\hat{\mathbf{y}}$
Obtain the initial support $\mathcal{T}$ from $\hat{\mathbf{y}}^{\text{init}}$
for *each view* $v \in \mathcal{V}$ **do**
  Project $\mathcal{T}$ into view v
  Compute R_v using Eq. 14
Select the top-ranked diverse views $\mathcal{V}_K$
$\mathcal{V}_q \leftarrow \emptyset$
for *each view* $v \in \mathcal{V}_K$ **do**
  $b_v \leftarrow \text{Verify}(q, \mathcal{Q}_q, v, \text{Proj}_v(\mathcal{T}))$
  if $b_v = 1$ **then**
    $M_v \leftarrow \text{SAM2}(v, \text{Proj}_v(\mathcal{T}))$
    $\mathcal{V}_q \leftarrow \mathcal{V}_q \cup \{(v, M_v)\}$
$\hat{\mathbf{y}} \leftarrow \hat{\mathbf{y}}^{\text{init}}$
if $|\mathcal{V}_q| \geq 2$ **then**
  for *each point* $p_i \in \mathcal{P}$ **do**
    Compute ν_{iv}, m_{iv}, and w_{iv} for $(v, M_v) \in \mathcal{V}_q$
    Compute n_i from reliable observations
    if $n_i \geq n_{\min}$ **then**
      Compute e_i using Eq. 10
      Update $\hat{y}_i$ using Eq. 15
return $\hat{\mathbf{y}}$

B Dataset and Implementation Details

The dataset split and evaluation protocol are described in Sec. 4.1. This section provides the data processing, training, and inference settings used in the reported experiments.

B.1 Dataset and Data Processing

We use the Faro scans, functional annotations, task descriptions, and posed RGB-D observations provided by SceneFun3D (Delitzas et al., 2024). The official crop masks are applied before constructing the working point cloud. Scenes are voxelized at $2\,\text{cm}$ resolution while maintaining a mapping to the original Faro points, which is used to restore predictions for refinement and evaluation. One training task with missing or invalid ground-truth annotation is excluded during preprocessing.

Dense visual features are extracted from the posed RGB observations using a frozen CLIP ViT-B/32 encoder (Radford et al., 2021). For each 3D point, image features at its valid depth-consistent projections are aggregated across observations according to Eq. 11. The image and text branches share the same CLIP embedding space, and the resulting language-aligned 3D features are cached per scene.

Functional Query Parsing. Each task is parsed once by a locally deployed Qwen3-8B model (Yang et al., 2025). We use deterministic decoding with thinking disabled and structured JSON output. Only the task instruction is provided; no scene geometry or ground-truth annotation is used. The parsed query is cached throughout training and inference. The prompt is shown below.

Functional query parsing prompt

Parse the SceneFun3D task into functional grounding semantics.
Identify the **action**, the **object whose state changes**, any **tool or remote**, and the **smallest physical part that must be touched**. For tasks involving "using" or "with" an object, ground the relevant control on that tool rather than the state-changing object. Use spatial relations to distinguish repeated objects.
When applicable, map the interaction to one of: rotate, key_press, tip_push, hook_pull, pinch_pull, hook_turn, foot_push, plug_in, unplug.

Return exactly one JSON object with: task, context_object, functional_object, affordance, functional_relation, spatial_constraint, and target_description.
The `functional_object` and `target_description` should refer to the **smallest touched region**.
Do not output 3D coordinates or additional text.
Task: [TASK]
JSON:

Core implementation and training settings are summarized in Table 4.

Table 4: Core implementation and training settings.

Setting	Value
Voxel size	$0.02\,\mathrm{m}$
Visual/text encoder	CLIP ViT-B/32
Task parser	Qwen3-8B (local)
Maximum local input points	16,384
Model dimension	512
Encoder layers / heads	8 / 16
Window size	128
Optimizer	AdamW
Batch size	4
Learning rate	2×10^{-4}
Warmup	1,500 steps
Training loss	Focal + Dice
Hardware	NVIDIA A100 40GB

B.2 Training Details

Only the LocalFunctionalGrounder in Sec. 3.3 is trained; the pretrained visual and text encoders remain frozen. Training uses the focal–Dice objective in Eq. 13, with each local input containing up to 16,384 points. To reduce the train–inference gap, training samples include both ground-truth-centered regions and regions produced by the same task-conditioned narrowing procedure used at inference time. Ground-truth-centered regions are used only for training supervision and are never used to construct candidate regions at inference time. Standard geometric and appearance augmentations are applied to the local inputs.

We train with AdamW using a batch size of 4 and an initial learning rate of 2×10^{-4}, followed by cosine decay after 1,500 warmup steps. Training takes approximately 150 GPU-hours on a single NVIDIA A100 40GB GPU, with a peak memory usage of approximately 17 GB.

B.3 Inference Details

At inference time, the structured query and cached language-aligned features are first used to construct the candidate region $\mathcal{P}_q$ following Sec. 3.2. The LocalFunctionalGrounder then predicts the initial point-wise functional scores within this region.

Fixed-Budget Local Grounding. The local grounder processes at most 16,384 points. If $\mathcal{P}_q$ exceeds this budget, points are subsampled while preserving high-response regions and spatial coverage, and predictions are propagated to the remaining points in $\mathcal{P}_q$ by nearest-neighbor assignment in 3D. Points outside $\mathcal{P}_q$ receive zero initial probability. Predictions are then mapped from the voxelized representation back to the original Faro points for multi-view refinement and evaluation. For the scene-narrowing analysis in Table 2, all variants use the same trained LocalFunctionalGrounder checkpoint and the same local grounding description; only the candidate region supplied to the grounder changes. The 16,384-point budget is fixed across variants, and the Whole Scene reference applies the same fixed-budget sampling procedure directly to the unpruned scene. Multi-view refinement is disabled for this analysis.

Table 5: Core scene-narrowing and inference settings of ActFun3D.

Setting	Value
Context / target / task weights	$0.35/0.45/0.20$
Retained semantic fraction	10%
Initial expansion radius	$0.50\,\mathrm{m}$
DBSCAN ϵ / min samples	$0.20\,\mathrm{m}$ / 8
Maximum local input points	16,384
Initial support threshold	0.50
View selection	Geometry-guided score + diversity
Maximum retrieved views	12
View verifier	`qwen3.8-flash` (API)
2D segmenter	SAM2
Minimum verified views	2
$\lambda_{\mathrm{3D}}/\lambda_{\mathrm{vis}}$	$0.6/0.4$
Refinement strategy	Suppression-only (no visual promotion)

View Retrieval and Verification. Candidate views are ranked using Eq. 14, followed by diversity-aware filtering, and at most 12 views are retained. This retrieval stage is entirely geometry-guided and does not use a VLM.

For the full ActFun3D model, the retained views are verified using `qwen3.8-flash` through an API. The RGB image contains an overlay of the projected 3D support, and the verifier receives the task together with the target description derived from the structured query. The prompt used in our experiments is:

View verification prompt

Inspect the **highlighted candidate area**. Decide whether it visually supports the **functional target**.
Task: [TASK]
Target: [TARGET DESCRIPTION]
Return exactly one JSON object with **relevant** (boolean), **confidence** (0–1), **target_description** (string), and **reason** (string).

2D Segmentation and Refinement. Verified views are segmented using SAM2 (Ravi et al., 2025), with the projected 3D support used as the spatial prompt. The resulting masks are projected back into 3D and aggregated using Eq. 10. Points that are not reliably visible in a view contribute no evidence, preventing occlusion from being interpreted as negative support. Refinement follows Eq. 15, with $\lambda_{\mathrm{3D}} = 0.6$ and $\lambda_{\mathrm{vis}} = 0.4$. At least two verified views are required before visual evidence is applied. Visual promotion is disabled, so refinement can suppress unsupported initial responses but cannot independently introduce new positive regions.

Runtime Measurement. Runtime in Table 3 is reported as the average wall-clock inference time per task over all 445 tasks in SceneFun3D *split0*, using the same view budget of $K = 12$. All variants share the same initial 3D prediction, so the reported differences reflect the additional refinement cost. VLM-based variants include the latency of API-based view verification.

B.4 Evaluation Protocol

All quantitative results are evaluated on SceneFun3D *split0* unless otherwise specified. For AP_{25} and AP_{50}, we follow the official SceneFun3D Task-2 evaluation protocol (Delitzas et al., 2024). Final point-wise probabilities are thresholded at 0.5 to obtain the predicted foreground support, which is partitioned into spatial components using 3D clustering with a $0.08\,\mathrm{m}$ neighborhood radius. Each component is exported as an individual functional instance, with its confidence computed as the mean final probability of its constituent points. Predictions are then restored to the point ordering of the original Faro scan before evaluation.

Table 6: Point-level prediction quality before and after multi-view evidence refinement on SceneFun3D *split0*. Both variants use the same point-level evaluation protocol.

Configuration	Precision ↑	Recall ↑	F1 ↑
Initial 3D	0.303	**0.425**	0.354
Full ActFun3D	**0.390**	0.417	**0.403**

Table 7: Comparison with reported UniFunc3D results on SceneFun3D *split0*. UniFunc3D results are taken from the original work and are not reproduced by us.

Method	Source	AP_{50} ↑	AP_{25} ↑	mIoU ↑
UniFunc3D-8B (Lin & Xu, 2026)	Reported	23.82	44.04	20.92
UniFunc3D-30B (Lin & Xu, 2026)	Reported	31.24	51.01	24.30
ActFun3D	Ours	30.78	41.53	22.96

For tasks associated with multiple functional elements, multiple predicted spatial components are retained as separate instances. The official evaluator is used without modification for all reported AP metrics. We additionally report mIoU as a point-level overlap measure. Other point-level metrics are used only for diagnostic analysis and do not replace the official AP evaluation.

B.5 Scene Narrowing Metrics

The SSR and TR metrics defined in Sec. 4.3 are computed in the same voxelized working-point space used for scene narrowing, before fixed-budget sampling by the local grounder. For tasks with multiple functional annotations, the target support $\mathcal{Y}_q$ is defined as the union of all valid referenced functional regions.

SSR depends only on the predicted candidate region, whereas ground truth is accessed only after narrowing to compute TR. Thus, neither candidate generation nor the narrowing threshold uses target annotations. The Whole Scene setting is the unpruned reference with $\mathrm{SSR} = \mathrm{TR} = 100\%$.

Beyond the mean TR of 94.2%, target retention remains concentrated at high values: the median is 97.5% and the 10th percentile is 84.5%. Only 8.3% (37/445) of tasks retain less than 80% of the target, and only 2.0% (9/445) fall below 60%. The minimum observed TR is 0%, corresponding to a complete narrowing failure; representative failure cases are discussed in Appendix D.

B.6 Refinement Diagnostics

To further characterize the conservative multi-view refinement in Sec. 3.4, Table 6 reports point-level precision, recall, and F1 before and after refinement on SceneFun3D *split0*. These metrics are used only for diagnostic analysis.

As shown in Table 6, refinement increases precision from 0.303 to 0.390 and F1 from 0.354 to 0.403, while recall decreases only slightly from 0.425 to 0.417. This behavior is consistent with the suppression-only design: verified multi-view evidence primarily removes unsupported responses rather than introducing new positive regions.

B.7 Baseline Evaluation and Reproduction

Baseline Evaluation. We follow the evaluation protocol in Sec. 4.1 for all baselines in the main comparison. Their outputs are converted to the SceneFun3D Task-2 format and evaluated with the same official evaluator used for ActFun3D. In particular, we re-evaluate AffordBot (Wang et al., 2025) with the official evaluator rather than relying on intermediate point-level metrics. This ensures a consistent evaluation protocol across all reproduced baselines.

Comparison with UniFunc3D. UniFunc3D (Lin & Xu, 2026) follows a different grounding strategy based on progressive spatial-temporal localization across visual observations. Its source code

Table 8: Additional ablations of the LocalFunctionalGrounder on SceneFun3D *split0* before multi-view refinement. The architecture tuple denotes feature dimension / encoder layers / self-attention heads / window size.

(a) Model Capacity			
Configuration	$\mathrm{AP}_{50}\uparrow$	$\mathrm{AP}_{25}\uparrow$	mIoU ↑
Lightweight $(256/4/8/64)$	21.7	25.5	18.0
Default $(512/8/16/128)$	**24.1**	**26.2**	**19.1**
(b) Training Region Strategy			
Training Region	$\mathrm{AP}_{50}\uparrow$	$\mathrm{AP}_{25}\uparrow$	mIoU ↑
GT-centered Only	20.7	23.4	17.3
Narrowing-region Only	22.8	24.9	18.3
GT-centered + Narrowing-region	**24.1**	**26.2**	**19.1**

was not publicly available at the time of submission, preventing reproduction under our evaluation pipeline. We therefore exclude it from the main baseline comparison and, for completeness, separately report the results provided by the original work on the same SceneFun3D *split0* in Table 7.

As shown in Table 7, UniFunc3D-30B reports higher results, particularly in AP_{25}, while the two methods are closer in AP_{50} and mIoU. UniFunc3D progressively searches visual observations for task-relevant evidence, whereas ActFun3D first restricts the 3D search space and performs point-level grounding within the narrowed region, using selected posed RGB-D views only for subsequent refinement. Since UniFunc3D cannot be evaluated within our controlled pipeline, its reported results are presented separately from the reproduced comparison.

C Additional Experimental Results

C.1 Additional Ablation Studies

We provide additional ablations on the LocalFunctionalGrounder and evidence-view retrieval, complementing the scene-narrowing and refinement analyses in Tables 2 and 3.

Local grounding capacity and training regions. Table 8(a) examines the effect of model capacity before multi-view refinement. Reducing the architecture from $512/8/16/128$ to $256/4/8/64$ decreases AP_{50} from 24.1 to 21.7, while the drops in AP_{25} and mIoU are smaller (0.7 and 1.1 points, respectively). This suggests that additional local modeling capacity is particularly useful for stricter spatial localization, while the lightweight variant retains much of the grounding performance. Together with Table 2, where structured task semantics retain 94.2% of the target within only 15.9% of the scene, these results show that performance does not rely solely on increasing grounder capacity: scene narrowing first provides a cleaner search domain, after which the local model resolves the fine-grained target.

Table 8(b) further studies the source of training regions. Using narrowing-generated regions improves AP_{50} from 20.7 to 22.8 over GT-centered training alone, indicating that exposure to inference-time candidate regions helps the grounder handle imperfect local proposals. Combining both sources further improves performance to $24.1/26.2/19.1$. The two training sources are therefore complementary: narrowing-generated regions reduce the train–inference gap, while GT-centered regions provide stable target-focused supervision. This supports the mixed sampling strategy used in Sec. 3.3.

Evidence-view retrieval. Table 9(a) studies view selection while keeping verification, 2D segmentation, and multi-view fusion fixed. Starting from the same initial 3D prediction, random views improve AP_{50} from 24.1 to 27.7, showing that additional visual observations already provide useful evidence. Uniform sampling further improves the result to $28.6/35.4/21.3$, while geometry-based ranking reaches $29.9/39.2/22.6$, indicating that views with stronger geometric support provide more reliable refinement signals. Diversity-aware selection further raises performance to $30.8/41.5/23.0$,

Table 9: Analysis of evidence-view retrieval on SceneFun3D *split0*. All variants use the same initial 3D prediction, view verification, 2D segmentation, and multi-view evidence fusion. Unless otherwise specified, the view budget is $K = 12$.

(a) View Retrieval Strategy			
Strategy	$\mathrm{AP}_{50}\uparrow$	$\mathrm{AP}_{25}\uparrow$	$\mathrm{mIoU}\uparrow$
Initial 3D	24.1	26.2	19.1
Random	27.7	33.3	20.8
Uniform Sampling	28.6	35.4	21.3
Geometry Score	29.9	39.2	22.6
Geometry Score + Diversity	**30.8**	**41.5**	**23.0**
(b) View Budget			
K	$\mathrm{AP}_{50}\uparrow$	$\mathrm{AP}_{25}\uparrow$	$\mathrm{mIoU}\uparrow$
3	28.1	34.6	21.9
6	29.4	37.3	21.8
12	**30.8**	<u>41.5</u>	**23.0**
24	<u>30.4</u>	**41.6**	<u>22.9</u>

showing the benefit of complementary rather than redundant viewpoints. This matches the full ActFun3D result in Table 1.

Table 9(b) varies the retrieval budget. From $K = 3$ to $K = 12$, AP_{50} increases from 28.1 to 30.8 and AP_{25} from 34.6 to 41.5, while mIoU remains relatively stable before reaching 23.0 at $K = 12$. Beyond this point, performance largely saturates: using 24 views slightly increases AP_{25} to 41.6, but reduces AP_{50} to 30.4 and mIoU to 22.9. We therefore use $K = 12$ as the default budget, capturing most useful complementary evidence without introducing increasingly redundant observations.

C.2 Additional Qualitative Results

Figure 5 extends the qualitative comparison in Figure 3 to additional tasks and baselines. A recurring difference is the granularity of the predicted region. Generic open-vocabulary methods frequently respond at the object or part level rather than at the small region that enables the task. For *Open the bottom cabinet drawer with the globe on top*, for example, OpenIns3D covers a large portion of the drawer front, whereas the functional target occupies only small interaction regions. A similar effect appears for the washing-machine task, where some predictions extend to larger appliance regions instead of concentrating on the controls. Functionality-oriented methods generally localize the relevant area more reliably, but can still produce incomplete, displaced, or overly extended responses. ActFun3D more consistently concentrates predictions around the interaction region.

The difference becomes more apparent when the functional target is small or surrounded by nearby distractors. In the two remote-control examples, the target occupies only a small fraction of the local scene and must be distinguished from nearby objects using task context. ActFun3D produces compact responses around the intended remote, while competing methods more often exhibit missing, displaced, or over-extended predictions. In contrast, the cabinet-door example is comparatively less ambiguous, and several functionality-oriented methods already identify the elongated interaction region reasonably well. This suggests that the qualitative advantage is most apparent under strong scale imbalance or multiple plausible candidates.

Overall, these observations are consistent with the quantitative results in Table 1. The improvement is not limited to coarse target recovery at AP_{25}; the more compact and spatially aligned predictions are also consistent with the gains in AP_{50} and mIoU. ActFun3D nevertheless remains imperfect, with some small or elongated regions showing fragmentation or extent mismatch. These residual errors are discussed in Appendix D.

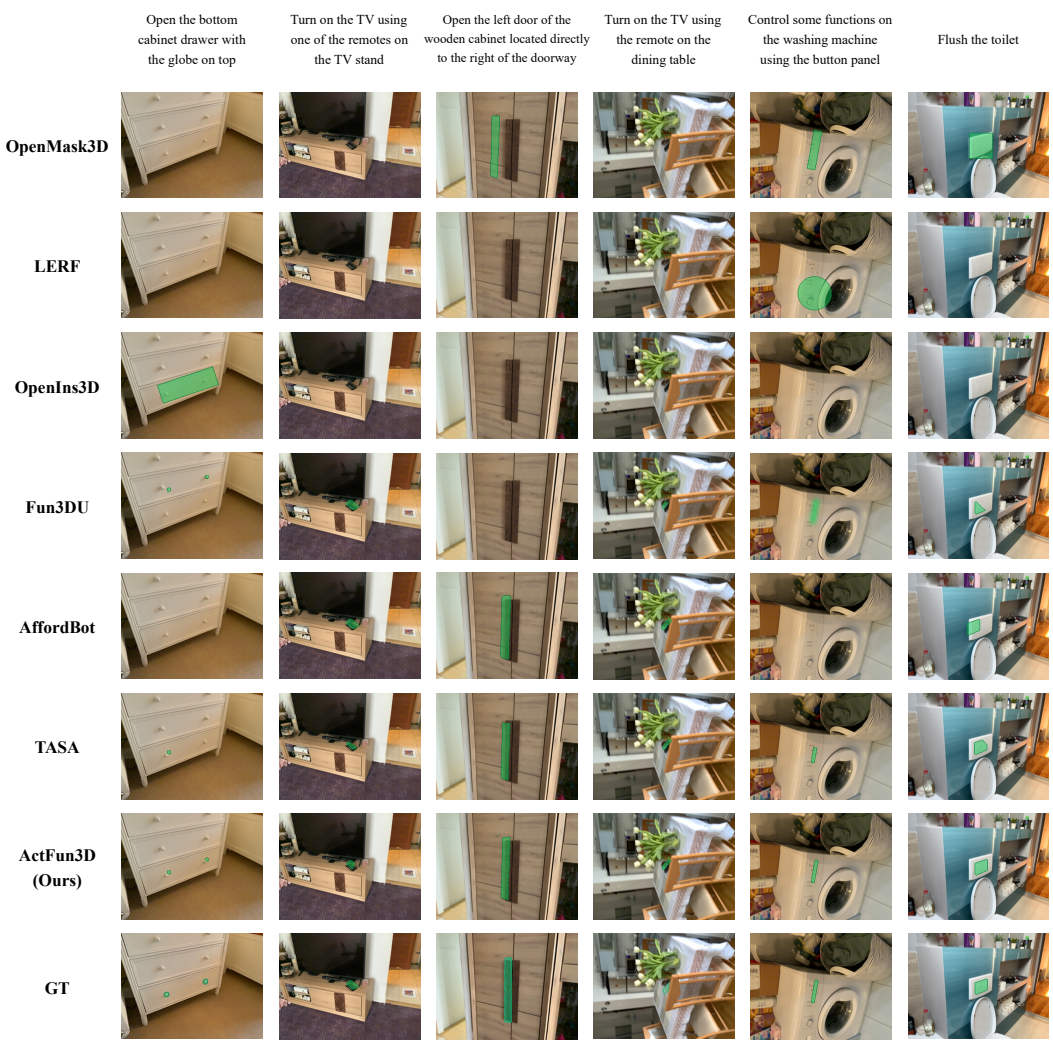


Figure 5: **Additional qualitative comparison on SceneFun3D *split0*.** Green overlays denote predicted functional regions. We compare ActFun3D with all baselines in Table 1; GT denotes ground truth.

C.3 Additional Scene Narrowing and Refinement Visualizations

Figure 6 further illustrates the coarse-to-fine behavior of ActFun3D. Scene narrowing preserves more than the final interaction region: it removes most irrelevant scene content while retaining the local object and surrounding context useful for subsequent grounding. For example, in *Open the right door of the closet near the window*, the narrowed region still covers a substantial portion of the closet while excluding most of the remaining scene. Similarly, for *Adjust the intensity of the plug-in air freshener*, nearby bathroom geometry is retained around the candidate device. This is consistent with Table 2, where the structured query preserves 94.2% of the target points while reducing the search space to 15.9% of the scene. The narrowing stage therefore acts as a high-recall search-space reduction step rather than a fine-grained segmentation module.

The subsequent refinement suppresses residual ambiguity within this reduced search space. In the closet-door example, the initial 3D prediction contains multiple responses around the door, whereas the final prediction becomes substantially more compact after incorporating verified visual evidence. A similar contraction is observed for the plug-in air freshener, where the broad initial response is reduced to a smaller functional region. In contrast, for *Open the bottom drawer of the cabinet with the TV on top*, the initial prediction is already compact and refinement largely preserves it.

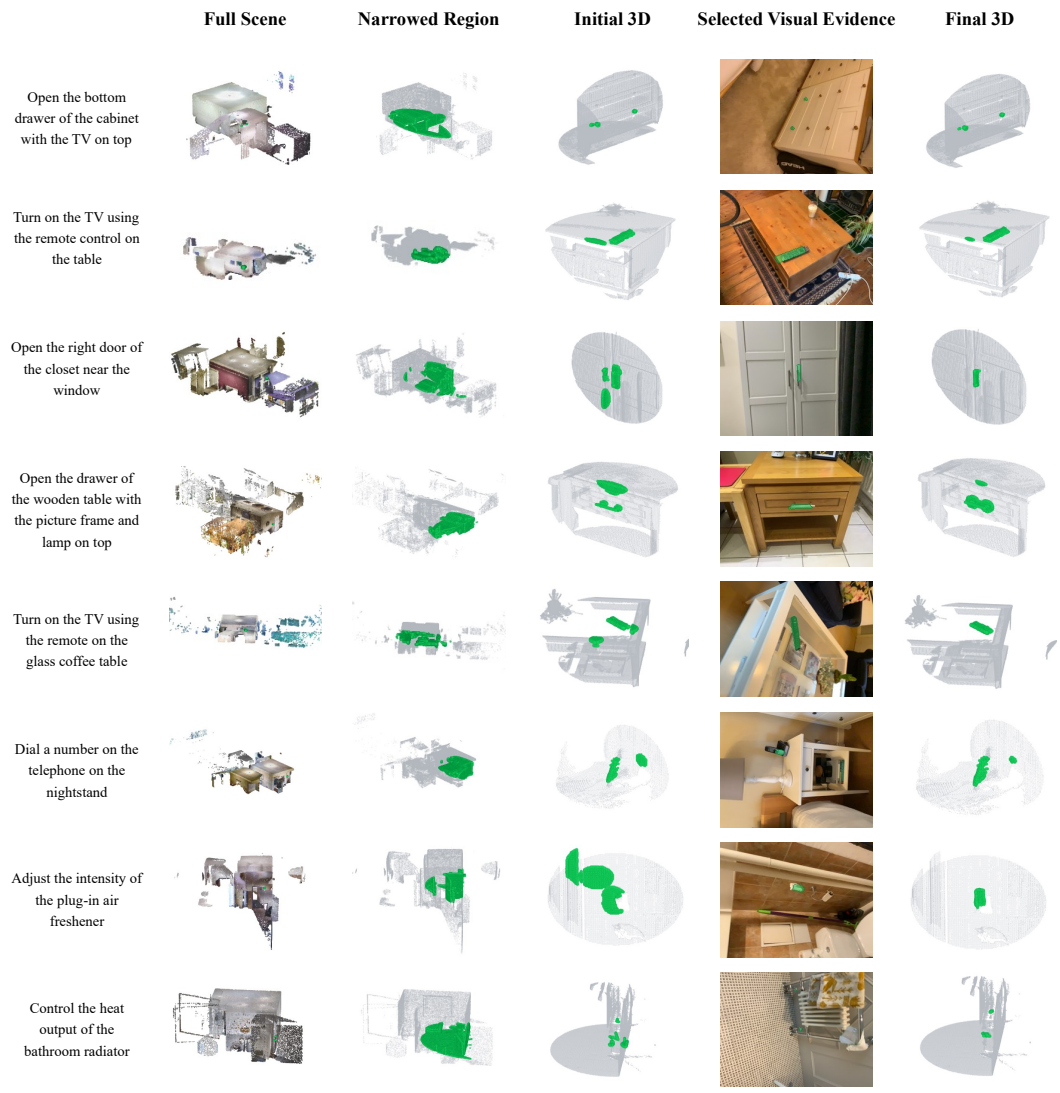


Figure 6: **Additional visualizations of task-conditioned scene narrowing and multi-view refinement.** From left to right, we show the full scene, the task-conditioned narrowed region, the initial 3D prediction, a representative selected visual evidence view, and the final refined 3D prediction.

These behaviors reflect the conservative design in Sec. 3.4: visual evidence suppresses unsupported responses while preserving reliable initial predictions rather than introducing new positive regions.

Figure 7 further isolates the retrieval and verification stages. Retrieved observations are not equally informative: some clearly expose the predicted functional region, whereas others are rejected because they provide insufficient task-relevant visual evidence. In the closet-door example, only a subset of retrieved views is retained, yet the verified evidence is sufficient to suppress additional initial responses and produce a more compact final prediction. For the plug-in air freshener, views that clearly expose the small interaction region are retained, while less informative observations are discarded before 3D evidence fusion. These examples complement the retrieval ablation in Table 9, qualitatively supporting the benefit of selecting informative and complementary observations.

The diagnostic examples also show that verification does not eliminate every error. In the wooden-table drawer example, several views are accepted and the prediction becomes more localized, but multiple spatial responses remain in the final result. Thus, refinement can substantially suppress unsupported predictions, but its effectiveness remains constrained by ambiguity in the initial 3D

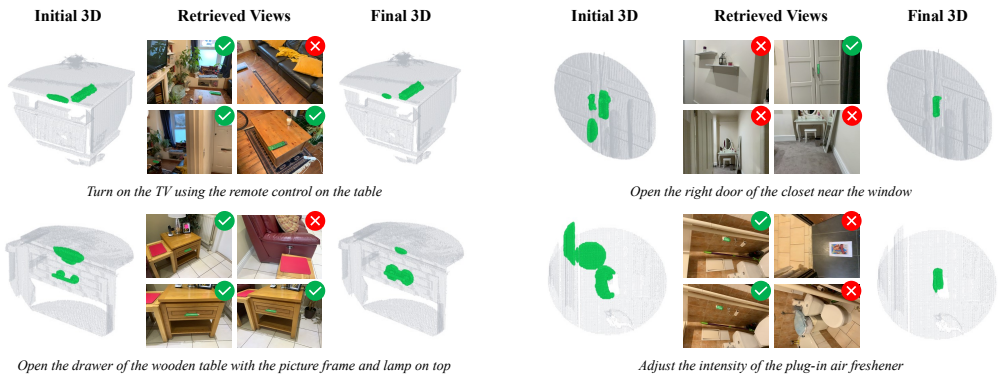


Figure 7: **Diagnostic visualization of evidence-view retrieval and verification.** For each task, we show the initial 3D prediction, a representative subset of the retrieved candidate views, and the final refined prediction. Green check marks indicate views accepted by task-aware verification, while red crosses indicate rejected views.

candidates and the evidence available in the retrieved views. We discuss these residual failure modes in Appendix D.

D Failure Cases and Limitations

Scene-narrowing failures. Although structured scene narrowing retains 94.2% of the target on average, failures remain in the tail of the distribution. As discussed in Appendix B.5, 8.3% (37/445) of tasks retain less than 80% of the target, 2.0% (9/445) fall below 60%, and the minimum observed TR is 0%. The latter corresponds to a complete narrowing failure, where the functional target is excluded from the candidate region. Such failures can occur when task cues are weak or ambiguous, or when several scene regions are similarly compatible with the structured query. Selecting a single dominant component can also discard valid target regions when a task involves multiple spatially separated functional elements. Because subsequent grounding operates within the retained region, target evidence removed at this stage cannot be recovered by later refinement.

Residual ambiguity. ActFun3D does not always fully resolve multiple plausible interaction regions. As shown in Fig. 7, for *Open the drawer of the wooden table with the picture frame and lamp on top*, refinement makes the prediction more compact but still leaves multiple spatial responses. Similar ambiguity can occur for small or repeated objects when multiple regions receive plausible task and visual evidence. View verification can suppress clearly unsupported candidates, but may remain insufficient when competing regions are visually and functionally similar. Small or elongated targets may also remain fragmented or differ from the ground-truth extent.

Dependence on initial 3D support. The refinement stage is deliberately conservative and therefore depends on the quality of the initial prediction. While this prevents noisy visual evidence from introducing unsupported positive regions, it also limits recovery when the initial grounding misses part of the functional target. For example, the bathroom-radiator case in Fig. 6 produces a relatively sparse final response. More generally, refinement can suppress false responses but cannot restore target regions absent from the initial 3D support, making errors from scene narrowing and local grounding difficult to correct downstream.

Observation and inference cost. ActFun3D assumes a fixed set of posed RGB-D observations, so refinement is limited by what is visible in the available views. Poor visibility, occlusion, or insufficiently discriminative observations can reduce the effectiveness of verification and evidence fusion. VLM-based verification also introduces substantial inference overhead: the full model requires 126.6 s/task compared with 16.3 s/task for the initial 3D prediction. Future work will investigate adaptive view acquisition, more efficient task-aware verification, and refinement mechanisms that can recover missing target support without sacrificing the stability of conservative fusion.