LEARNING NOT TO DISCARD PREMATURELY: TRAJECTORY LEARNING FOR INTERACTIVE 3D VISUAL GROUNDING

Anonymous authors
Paper under double-blind review

ABSTRACT

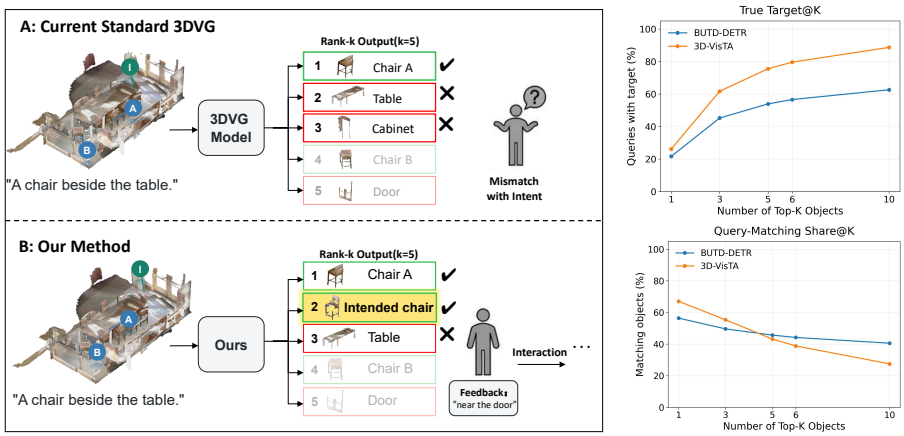
3D visual grounding aims to localize the query-matched target object in a 3D scene. In real-world scenarios such as human–robot interaction, human queries are often ambiguous and may be compatible with multiple candidate objects, necessitating iterative interaction to identify the intended target. However, as the interaction progresses, candidate filtering based solely on current query–object matching often fails to explicitly account for its consequences for subsequent interaction. We observe that interaction trajectories reveal which early filtering decisions successfully retain the target until disambiguating evidence arrives, and which make later clarification ineffective. Building on this observation, we formulate interactive 3D visual grounding as sequential candidate refinement, and propose a trajectory learning framework that uses ordered candidate transitions, subsequent feedback, and final outcomes to optimize early candidate filtering decisions. Through trajectory-level preference optimization, the framework learns to retain plausible candidates until sufficient clarification evidence arrives. To effectively propagate trajectory-level supervisory signals to the early candidate generation, we introduce a trajectory-aware adapter that integrates interaction history and candidate belief to produce residual updates for candidate representations and grounding scores. We evaluate our framework on AmbiRefer3D, an interactive 3D visual grounding dataset, and construct a more challenging subset requiring multiple rounds of clarification to uniquely identify the target. Experimental results demonstrate that our method improves candidate retention and downstream interaction performance, outperforming the direct-training baseline by an average of 20.56% in TS and 3.66% in Acc across grounding backbones. These results highlight that learning how to avoid prematurely discarding candidates is crucial for achieving reliable interactive 3D visual grounding.

1 INTRODUCTION

3D visual grounding (3DVG) aims to localize objects in a 3D scene from natural language referring expressions (Chen et al., 2020; Achlioptas et al., 2020), supporting embodied applications such as human-robot collaboration (Wang et al., 2026a; Kato et al., 2026), object manipulation (Huang et al., 2025; Lu et al., 2026), and navigation (Xu et al., 2026; Luo et al., 2026). Recent work has advanced 3DVG in several directions, including modeling contextual and spatial relations (Yang et al., 2023; Wang et al., 2025; Sun et al., 2026), aligning fine-grained linguistic and visual features (Wu et al., 2023; Xu et al., 2024a; Gu et al., 2025), and grounding multiple query-compatible objects (Zhang et al., 2023; Zheng et al., 2025; Li et al., 2026). With sufficiently informative descriptions, these methods can accurately distinguish and localize the target object.

However, in real-world scenarios such as human-robot interaction, human queries are often ambiguous and necessitate iterative interaction to identify the intended object. As the interaction progresses, subsequent clarification provides additional information, enabling the model to progressively narrow the set of candidates. However, this process highly depends on the quality of the candidate set in each round. As illustrated in Fig. 1(a), the query "A chair beside the table" may match multiple candidates, making immediate matching scores alone insufficient to identify the user's intended target. Moreover, simply expanding the candidate budget does not solve the problem. As shown in

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.



(a) (b)

Figure 1: (a) With an ambiguous query, a ranked matching object may not be the intended target. Our approach retains plausible candidates for subsequent interaction rather than discarding the target prematurely. (b) True Target@K and Query-Matching Share@K results for BUTD-DETR and 3D-VisTA under different candidate budgets.

Fig. 1(b), an increase in target coverage is accompanied by a decrease in the proportion of relevant candidate objects. As the interaction progresses, candidate object filtering based solely on the current query–object compatibility often fails to explicitly account for how each decision may affect subsequent disambiguation. Under irreversible candidate pruning, prematurely discarding the target object makes subsequent clarification interactions ineffective. Therefore, the challenge lies in how to retain plausible candidates until sufficient evidence justifies discarding them.

We observe that interaction trajectories reveal which early filtering decisions retain the target until disambiguating evidence arrives, and which make subsequent clarification ineffective. Each trajectory records the initial ambiguous query, the candidates matching the updated query at each round, clarification constraints, human feedback, and the target object. The candidates that match the updated query indicate which objects should be retained, while candidates changing following feedback indicates when candidates can be discarded. Final results reveal whether the interaction preserves the target and resolves ambiguity, or stops too early.

Building on this observation, we formulate interactive 3D visual grounding as a sequential candidate refinement process, and introduce trajectory-level preference optimization for early candidate filtering. Specifically, we construct a comparison of preferred and rejected trajectories based on target retention, candidate reduction, and interaction cost. Learning from these comparisons encourages filtering decisions to preserve plausible candidates for subsequent clarification, rather than prematurely discarding them. To effectively propagate trajectory-level supervisory signals for early candidate generation, we propose a trajectory-aware adapter that combines query representations, feedback, and changes in the candidate distribution to refine candidate representations and grounding scores. Therefore, the proposed framework can encourage the model to retain plausible candidates even with incomplete evidence and progressively optimize the candidate set during the interaction process, as illustrated in Figure 1(a).

We evaluate the proposed framework on AmbiRefer3D (Zhu et al., 2026), an interactive 3D visual grounding dataset. To better evaluate long-horizon disambiguation capabilities, we additionally construct a more challenging subset that requires multiple rounds of clarification to uniquely identify the target. These examples are manually annotated with initial ambiguous queries, and their scenes are modified as needed so that attribute or relationship clarification can not uniquely identify the target in early rounds, thus providing an effective evaluation of candidate retention and long-horizon disambiguation capabilities. Experiments across multiple grounding backbones demonstrate that our method improves candidate retention and final grounding accuracy, with average gains of 20.56% and 3.66% in TS and Acc, respectively, compared to the direct-training baseline.

The contributions of this work are summarized as follows:

- We formulate interactive 3D visual grounding as a sequential candidate refinement process, and propose a trajectory learning framework to retain plausible candidates until sufficient clarification evidence arrives. This provides a solution to reduce the risk of discarding the intended target in real-world interactive scenarios.
- We propose a trajectory-aware adapter that propagates trajectory-level supervisory signals to generate candidate objects, enabling joint effective optimization with subsequent interaction.
- We construct a more challenging multi-round subset of AmbiRefer3D and evaluate our framework on multiple grounding backbones, demonstrating its effectiveness and highlighting the importance of learning how to avoid prematurely discarding candidates for achieving reliable interactive 3D visual grounding.

2 Related Work

Advances in 3D visual grounding have improved object localization from natural-language descriptions through contextual reasoning, cross-modal alignment, and multi-view modeling (Chen et al., 2020; Achlioptas et al., 2020; Zhao et al., 2021; Wu et al., 2023; Guo et al., 2023). Further work extends grounding to multiple query-matched objects (Zhang et al., 2023), improves efficiency through text-guided voxel pruning (Guo et al., 2025), and incorporates rendered views and spatial context (Li et al., 2025; Dinh et al., 2026). These advances primarily concern grounding from a given description. Our work focuses on interactive 3D visual grounding with referential ambiguity, accounting for how early filtering decisions affect subsequent disambiguation. We therefore discuss two closely related lines of research: interactive visual grounding and trajectory learning in 3D environment.

2.1 Interactive Visual Grounding

Interactive visual grounding addresses referential ambiguity by allowing a model to actively acquire additional information from a user. INGRESS (Shridhar & Hsu, 2018) identifies candidate objects in images and generates clarification questions using visual descriptions and pairwise relations. InViG (Zhang et al., 2024) introduces large-scale open-ended disambiguation dialogues, while TiO (Xu et al., 2024b) unifies visual dialogue and grounding for human-robot interaction. These works primarily study interactive disambiguation in 2D images. Extending to 3D scenes, ScanERU (Lu et al., 2024) combines language with synthetic gestural cues for embodied reference understanding, relying on additional gesture input. AmbiRefer3D (Zhu et al., 2026) instead studies language-based multi-round clarification for referential ambiguity in point-cloud scenes. Although existing interactive 3D visual grounding methods resolve referential ambiguity, a further problem is to account for how early candidate filtering affects subsequent disambiguation. To address this problem, we learn from interaction trajectories to reduce the risk of prematurely discarding the target and retain plausible candidates until sufficient evidence arrives.

2.2 Trajectory Learning in 3D Environments

Trajectory learning models temporal dependencies in sequential experiences to support representation learning and decision-making. In 3D environments, prior work encodes observation-action histories for navigation (Chen et al., 2021), learns view-selection policies to reduce uncertainty about unobserved content (Jayaraman & Grauman, 2018), and optimizes viewpoint selection through coverage-based rewards (Chen et al., 2024). Recent methods align visual histories with instruction progress (Wang et al., 2026b), compress action-pose sequences into trajectory memories (Jonna et al., 2026), and improve navigation policies through time-decayed rewards (Qi et al., 2025) and horizon-adaptive advantage estimation (Hu et al., 2026). In interactive 3D visual grounding, trajectories instead track how clarification feedback changes candidate sets and whether early filtering preserves the target until disambiguating evidence arrives. We exploit these structured transitions and final outcomes to learn candidate retention, using a trajectory-aware adapter to refine candidate representations and grounding scores from interaction history.

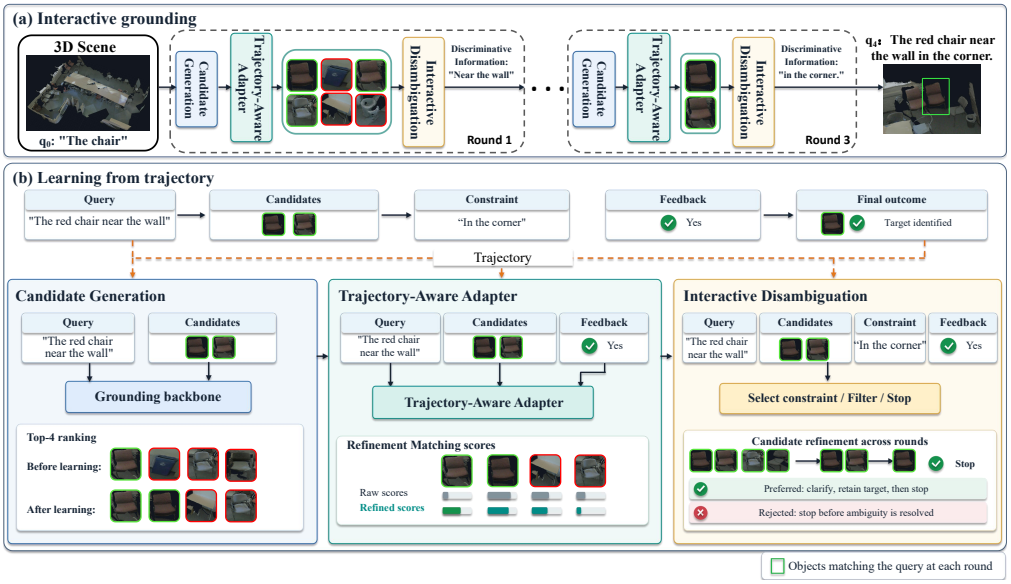


Figure 2: Overview of our framework. Interactive grounding progressively refines an initial candidate set, while training uses query, candidates matching the query at each round, clarification feedback, and final outcomes to retain plausible candidates until sufficient evidence justifies discarding them.

3 METHOD

3.1 OVERVIEW

To reduce the risk of discarding the intended target under insufficient evidence, we formulate interactive 3D visual grounding as sequential candidate refinement, retaining plausible candidates while ambiguity persists and pruning them only when sufficient clarification evidence becomes available. Our framework comprises three components: a candidate generation module, a trajectory-aware adapter, and an interactive disambiguation module. Specifically, the candidate generation module produces candidate proposals, candidate representations, and grounding scores. Then, the trajectory-aware adapter integrates these outputs with candidate belief and available interaction history from previous round to produce updates to candidate representations and grounding scores. Based on the refined representations and scores, the interactive disambiguation module selects clarification constraints and uses the feedback to update the candidate set and candidate belief and determine whether to stop. This process progressively refines the candidate set until localizing the target object or the maximum number of rounds is reached. Figure 2 illustrates the overall framework.

3.2 CANDIDATE GENERATION

Given a scene point cloud P and an ambiguous query q^0, interactive 3D visual grounding aims to identify the intended target $o^\star$ through iterative clarification. An interaction trajectory begins with q^0 and an initial candidate set $\mathcal{C}^0$, followed by T clarification transitions. At each round t, the model selects a clarification constraint a^t and receives feedback u^{t+1}, which is incorporated into the cumulative query q^{t+1} and used to refine the candidate set to $\mathcal{C}^{t+1}$, for $t \in \{0, \dots, T-1\}$.

In the candidate generation module, we use existing 3D visual grounding models to process the scene point cloud P and the cumulative query q^t, producing candidate proposals and representations and grounding scores for subsequent refinement:

$$(\mathbf{B}, \mathbf{V}, \mathbf{Z}^t, \boldsymbol{\ell}^t, \mathbf{e}^t) = \mathcal{G}(P, q^t), \tag{1}$$

where $\mathcal{G}$ denotes the grounding backbone, $\mathbf{B}$ denotes the candidate boxes, and $\mathbf{V}$ denotes the candidate representations. $\mathbf{Z}^t$ contains cross-modal features, which are concatenated with the adapter-refined candidate representations as input to the interactive disambiguation module. $\boldsymbol{\ell}^t$ is the grounding score, and $\mathbf{e}^t$ is the embedding of the cumulative query q^t.

3.3 TRAJECTORY-AWARE ADAPTER

In addition to the outputs from candidate generation, the adapter receives the retention weights $\boldsymbol{\alpha}^{t-1}$, the candidate belief $\boldsymbol{\omega}^{t-1}$, and a history state $\mathbf{h}^{t-1}$ from the previous round. Each retention weight $\alpha_i^{t-1} \in [0,1]$ represents the degree to which candidate i is retained, with larger values favoring retention. ω_i^{t-1} is obtained by normalizing the weighted refined scores across valid candidates in the previous round. Thus, the adapter produces refined candidate representations and grounding logits, together with an updated candidate distribution and history state:

$$(\widetilde{\mathbf{V}}^t, \widetilde{\boldsymbol{\ell}}^t, \boldsymbol{\omega}^t, \mathbf{h}^t) = \mathcal{A}(\mathbf{V}, \mathbf{B}, \boldsymbol{\ell}^t, \mathbf{e}^t, \boldsymbol{\alpha}^{t-1}, \boldsymbol{\omega}^{t-1}, \mathbf{h}^{t-1}, u^t), \tag{2}$$

where $\mathcal{A}$ denotes the adapter,$\widetilde{\mathbf{V}}^t$ indicates the refined candidate representations, $\widetilde{\boldsymbol{\ell}}^t$ indicates the refined grounding scores, and u^t is the feedback. At initialization, $\mathbf{h}^{-1} = \mathbf{0}$, $u^0 = 0$, and $\alpha_i^{-1} = m_i$, where m_i is a validity mask that equals 1 for a generated candidate box and 0 for a padded position.

The adapter first combines previous retention weight with current grounding score by computing $\alpha_i^{t-1}\sigma(\ell_i^t)$ for each candidate, where $\sigma(\cdot)$ denotes the sigmoid function. Normalizing these values across valid candidates yields the initial candidate belief distribution $\boldsymbol{\omega}_{\text{before}}^t$. The adapter uses $\boldsymbol{\omega}_{\text{before}}^t$ to aggregate the projected candidate representations for the history update. To capture geometric context for each candidate, for each candidate i, the adapter aggregates its encoded spatial relations with candidates j, weighted by ω_{before}^t, to obtain its geometric context.

To enable the model to track how candidates change across interaction rounds, we encode the feedback sign u^t together with three statistics: the entropy of $\boldsymbol{\omega}_{\text{before}}^t$, summed absolute difference from $\boldsymbol{\omega}_{\text{before}}^{t-1}$, and the mean retention weight. A GRU takes this encoding, the query embedding, and the aggregated feature as input and updates the history state from $\mathbf{h}^{t-1}$ to $\mathbf{h}^t$. For each candidate, its projected representation and geometric context are concatenated with $\mathbf{h}^t$ and fed into two prediction heads to produce bounded feature correction $\Delta\mathbf{v}_i^t$ and score correction d_i^t. These corrections are applied through residual updates, formulated by

$$\widetilde{\mathbf{v}}_i^t = \mathbf{v}_i + \lambda_r \Delta\mathbf{v}_i^t,\ \widetilde{\ell}_i^t = \ell_i^t + \lambda_r\big([d_i^t]_+ + g^t[d_i^t]_-\big), \tag{3}$$

where $\widetilde{\mathbf{v}}_i^t$ and $\widetilde{\ell}_i^t$ denote the refined candidate representation and grounding logit, respectively. λ_r is a learned residual scale. The operators $[x]_+ = \max(x, 0)$ and $[x]_- = \min(x, 0)$ separate positive and negative score corrections. The gate g^t modulates negative score corrections to reduce excessive suppression of plausible candidates. The refined candidate belief distribution $\boldsymbol{\omega}^t$ is obtained by normalizing $\alpha_i^{t-1}\sigma(\widetilde{\ell}_i^t)$ across valid candidates.

During trajectory training, we use a differentiable update to carry retention weights across rounds:

$$\alpha_i^t = \alpha_i^{t-1}\sigma\left(\frac{\widetilde{\ell}_i^t - \text{logit}(\eta_1)}{\tau_a}\right), \tag{4}$$

where η_1 is the filtering threshold, $\text{logit}(\cdot)$ denotes the logit function, and τ_a controls the smoothness of the update. To construct the initial candidate set for interactive disambiguation, we retain candidates whose refined matching scores exceed the filtering threshold as

$$\mathcal{C}^0 = \{i \mid m_i = 1,\ \sigma(\widetilde{\ell}_i^0) \geq \eta_1\}. \tag{5}$$

The selected candidates, together with their refined representations and grounding scores, are passed to the interactive disambiguation module. Details of adapter architecture, geometric encoding, history-state updates, and gating are provided in the Appendix A.

3.4 INTERACTIVE DISAMBIGUATION

To resolve the remaining ambiguity, the interactive disambiguation module selects clarification constraints and progressively refines the candidate set. The state s^t combines the refined representations, grounding scores, current candidates, and constraint context. The module selects a constraint a^t, receives the feedback u^{t+1}, and predicts candidate retention logits and a stopping logit conditioned on both the constraint and the feedback:

$$\boldsymbol{\pi}^t = \mathcal{D}_\psi^{\text{sel}}(s^t),\ (\mathbf{k}^t, z^t) = \mathcal{D}_\psi^{\text{filter}}(s^t, a^t, u^{t+1}). \tag{6}$$

Here, $\mathcal{D}_\psi^{\text{sel}}(\cdot)$ predicts a distribution $\boldsymbol{\pi}^t$ over the available clarification constraints $\mathcal{Q}$, from which $a^t = \arg\max_{a\in\mathcal{Q}} \pi_a^t$ is selected. Given the current state, selected constraint, and feedback u^{t+1}, $\mathcal{D}_\psi^{\text{filter}}(\cdot)$ outputs candidate-retention logits $\mathbf{k}^t$ and a stopping logit z^t. The feedback indicates whether the intended target satisfies the selected constraint.

To reduce the risk of prematurely discarding the target object, we lower the filtering threshold when constraint-selection confidence is low. We also accumulate scores to combine retention evidence across rounds:

$$\eta_2^t = \frac{\eta_2}{2}\left(1 + \max_{a\in\mathcal{Q}} \pi_a^t\right),\ L_i^{t+1} = L_i^t + \log\sigma(k_i^t), \tag{7}$$

where, η_2 is the filtering threshold, and L_i^t is the cumulative score of candidate i. We initialize $L_i^0 = \log p_i^0$, where $\mathbf{p}^0$ is the initial candidate belief distribution normalized over $\mathcal{C}^0$. Candidates in $\mathcal{C}^t$ satisfying $\sigma(k_i^t) \geq \eta_2^t$ are retained to form $\mathcal{C}^{t+1}$.

The interaction continues until the stopping criterion is met, or the maximum number of rounds is reached. At the final round T, the prediction is

$$\widehat{i} = \arg\max_{i\in\mathcal{C}^T} L_i^T, \qquad \widehat{o} = o_{\widehat{i}}. \tag{8}$$

Detailed interactive disambiguation architecture state construction, constraint schedules, and filtering and stopping rules are provided in Appendix A.2.

3.5 Training Objective

We jointly optimize the trainable components using grounding supervision, trajectory retention supervision, supervised interaction losses, and trajectory-level preferences:

$$\mathcal{L} = \mathcal{L}_{\text{cg}} + \lambda_{\text{ret}}\mathcal{L}_{\text{ret}} + \lambda_{\text{int}}\big(\mathcal{L}_{\text{sup}} + \lambda_{\text{pref}}\mathcal{L}_{\text{ipref}}\big). \tag{9}$$

Here, $\mathcal{L}_{\text{cg}}$ preserves the original grounding objectives, while the retention loss $\mathcal{L}_{\text{ret}}$ encourages target retention across rounds and supervises changes in the candidate set matching the accumulated query. The interaction loss $\mathcal{L}_{\text{sup}}$ trains constraint selection, retention of candidates matching the selected constraint, and stopping decisions. For preference supervision, we construct preferred and rejected comparisons from recorded interaction trajectories. The trajectory-preference loss $\mathcal{L}_{\text{ipref}}$ uses a DPO (Rafailov et al., 2023)-inspired objective to increase the score difference between preferred and rejected alternatives relative to that of a frozen reference interaction model. This encourages clarification decisions that support subsequent disambiguation. The coefficients λ_{ret}, λ_{int}, and λ_{pref} balance the objectives. More details of loss construction are provided in the Appendix A.3.

4 Experiments

4.1 Experimental Setup

Dataset. We use the Mixture type of AmbiRefer3D (Zhu et al., 2026), an interactive 3D visual grounding dataset. The training split contains 448 scenes and 10,707 ambiguous referring expressions, from which we construct 20,207 ordered interaction trajectories. The test split contains 113 scenes and 2,768 referring expressions. We additionally construct a more challenging subset containing 220 interaction trajectories. Each trajectory requires multiple rounds of clarification to uniquely identify the intended target. Detailed construction procedures are provided in Appendix C.

Evaluation metrics. We report target survival (TS), final grounding accuracy (Acc), Success@K, and the average number of interaction rounds (AvgR). TS measures whether the intended target (IoU$\geq$ 0.5) survives candidate generation. Success@K is the percentage of queries correctly grounded and terminated within K rounds, while AvgR denotes the average number of rounds for successful grounding.

Implementation details. We use M3DRefCLIP (Zhang et al., 2023), BUTD-DETR (Jain et al., 2022), 3D-VisTA (Zhu et al., 2023), and TSP3D (Guo et al., 2025) as candidate generation backbones in our framework, each initialized from its respective pretrained checkpoint. The trajectory-aware adapter uses a GRU with a 128-dimensional hidden state and two MLP heads for residual

Table 1: Comparison on the AmbiRefer3D test set.

Training scheme	Backbone	TS	Acc	S@1	S@3	S@6	AvgR
Direct	M3DRefCLIP	58.26	38.74	26.52	32.25	38.74	3.43
	BUTD-DETR	64.69	26.92	11.79	19.26	26.92	3.29
	3D-VisTA	74.16	30.28	16.42	25.73	30.28	3.26
	TSP3D	80.43	28.41	13.20	19.21	28.41	3.04
Shuffled pairs	M3DRefCLIP	79.54	39.11	28.13	35.52	39.11	3.16
	BUTD-DETR	76.73	24.31	13.24	17.52	24.31	3.24
	3D-VisTA	91.61	37.55	26.84	30.07	37.55	2.89
	TSP3D	84.47	28.67	15.45	19.82	28.67	3.15
Ordered trajectory	M3DRefCLIP	87.02	44.37	29.89	37.56	44.37	2.88
	BUTD-DETR	83.93	26.99	14.12	23.42	26.99	2.94
	3D-VisTA	97.28	37.96	24.64	28.70	37.96	2.75
	TSP3D	91.54	29.68	18.43	24.33	29.68	2.67

feature and score refinement. Interactive disambiguation uses a four-layer Transformer encoder with four attention heads and a hidden dimension of 256 for constraint selection. The filtering module predicts candidate retention logits through a cross-attention head and a stopping logit through an MLP head. We adopt the AdamW optimizer with module-specific learning rates. The initial learning rates for trainable detector backbone parameters, detector heads, grounding modules, the trajectory-aware adapter, and interactive disambiguation module are 2×10^{-7}, 5×10^{-7}, 1×10^{-6}, 2×10^{-5}, and 2×10^{-4}, respectively. We set the initial candidate generation threshold, disambiguation filtering threshold, and stopping threshold to 0.01, 0.4, and 0.5, respectively. We set $\lambda_{\text{ret}} = 0.2$, $\lambda_{\text{int}} = 1.0$, and $\lambda_{\text{pref}} = 0.5$. The maximum number of interaction rounds is set to 6. Additional architectural details and backbone-specific training schedules are provided in Appendix A.

4.2 Comparison Results

Main results. To evaluate whether trajectory-level learning improves candidate retention and downstream disambiguation, we compare three training schemes on AmbiRefer3D across four backbones. **Direct** freezes the pretrained backbone and trains the interactive disambiguation module using single-step supervision, following the original AmbiRefer3D interaction setting. **Shuffled pairs** treats each interaction round as an independent training example and shuffles these examples, removing cross-round dependencies. **Ordered trajectory** jointly optimizes candidate generation and disambiguation using ordered candidate transitions and trajectory-level preference learning.

As shown in Table 1, Shuffled pairs improves TS and Acc over Direct by 13.70% and 1.32 % in average, respectively. This indicates that directly supervising individual interaction round offers partial benefits, even without modeling their sequential dependencies. However, BUTD-DETR exhibits a 2.61 % decrease in Acc despite improved TS, showing that additional observation-level supervision does not necessarily translate into better downstream grounding performance. This contrast motivates learning how candidate retention decisions affect subsequent disambiguation, rather than treating interaction rounds solely as independent training examples.

Ordered trajectory further improves average TS and Acc over Shuffled pairs by 6.86% and 2.34% respectively, with gains in both metrics across all four backbones. These results support the value of trajectory-level supervision beyond using the same observations independently. In particular, the consistent TS improvements suggest that learning from candidate transitions and subsequent interaction outcomes helps shape initial candidate retention, while the accompanying Acc gains show that these improvements benefit downstream disambiguation. Moreover, AvgR decreases across all four backbones, suggesting that our framework helps models learn to follow more efficient interaction trajectories with fewer clarification rounds.

Overall, these results demonstrate that our framework improves early target retention, final grounding accuracy and interaction efficiency across different grounding backbones, highlighting the effectiveness of trajectory-level learning for interactive 3D visual grounding. Also, these findings highlight the importance of learning how to avoid prematurely discarding candidates for reliable interactive 3D visual grounding.

Table 2: Comparison on the AmbiRefer3D challenging subset.

Training scheme	Backbone	TS	Acc	S@1	S@3	S@6	AvgR
Direct	M3DRefCLIP	54.74	20.00	7.41	16.58	20.00	3.61
	BUTD-DETR	51.82	15.91	5.78	11.55	15.91	3.72
	3D-VisTA	51.82	17.73	10.78	14.21	17.73	3.44
	TSP3D	66.82	15.91	7.43	10.54	15.91	2.75
Shuffled pairs	M3DRefCLIP	68.11	21.05	7.86	15.37	21.05	3.47
	BUTD-DETR	56.36	15.55	6.58	13.79	15.55	3.68
	3D-VisTA	80.91	17.18	10.54	14.26	17.18	3.58
	TSP3D	65.45	16.39	8.54	13.30	16.39	2.89
Ordered trajectory	M3DRefCLIP	76.36	25.73	11.52	17.56	25.73	3.38
	BUTD-DETR	58.73	16.62	7.83	14.52	16.62	3.24
	3D-VisTA	88.64	19.09	12.26	15.36	19.09	3.72
	TSP3D	70.00	17.27	9.06	13.67	17.27	2.44

Table 3: Component ablation with M3DRefCLIP.

Adapter	Ret.	Gate	Pref	TS	Acc	AvgR
				65.43	40.18	3.48
✓				68.16	40.91	3.32
	✓			72.80	40.52	3.69
✓	✓			78.54	40.86	3.42
✓		✓		76.98	40.45	3.35
✓	✓	✓		80.44	41.54	3.23
	✓		✓	81.40	42.33	2.81
✓	✓	✓	✓	**87.02**	**44.37**	**2.88**

Challenging Subset. As shown in Table 2, we further evaluate three training schemes on the more challenging subset. Compared with Direct, Shuffled pairs increases backbone-averaged TS by 11.41%, while Acc remains largely unchanged. In contrast, Ordered trajectory improves average TS and Acc over Shuffled pairs by 5.73% and 2.14%, respectively, supporting the effectiveness of trajectory-level learning in difficult disambiguation scenarios. These results support our view that improved initial retention alone is insufficient. Learning to retain plausible candidates and progressively refine candidates across rounds is important for converting retention gains into successful grounding.

4.3 Ablation Studies

To evaluate the contribution of each component, we conduct ablation experiments on the trajectory-aware adapter, retention supervision, gate, and trajectory preference learning. As shown in Table 3, adding the adapter alone yields only limited improvements of 2.73% in TS and 0.73% in Acc. The adapter provides the capacity to incorporate interaction history, but this capacity alone does not ensure target-preserving predictions. Retention supervision alone improves TS by 7.37%, and combining it with the adapter further increases TS from 72.80 to 78.54. This suggests that explicit retention supervision helps the adapter use its additional capacity to preserve plausible targets. Adding the risk gate to this combination further improves TS and Acc by 1.90 and 0.68%, respectively. Finally, trajectory preference optimization provides additional gains of 6.58% in TS and 2.83% in Acc, while reducing AvgR from 3.23 to 2.88. These results demonstrate the complementary contributions of our components, providing a promising approach to preserving plausible candidates and progressively resolving ambiguity in real-world interaction.

4.4 Additional Analyses

Is ordered trajectory information necessary? We compare ordered trajectory training with shuffle round order and shuffle feedback. As shown in Table 4, shuffling the round order reduces TS from 87.02 to 79.21 and Acc from 44.37 to 40.30, while increasing AvgR from 2.88 to 3.48. This suggests that target retention alone is insufficient. Preserving the temporal structure of candidate transitions

Table 4: Ablation of trajectory information with M3DRefCLIP.

	TS	Acc	S@1	S@3	S@6	AvgR
Ordered trajectory	**87.02**	**44.37**	**29.89**	**37.56**	**44.37**	**2.88**
Shuffle round order	79.21	40.30	26.19	32.14	40.30	3.48
Shuffle feedback	71.58	23.15	14.52	19.30	23.15	3.24



Figure 3: Comparison of target coverage and query matching candidate share under different number of Top-K with M3DRefCLIP.

is also important for accurate and efficient disambiguation. Shuffling feedback causes larger degradation, reducing TS and Acc by 15.44 and 21.22%, respectively, highlighting the importance of aligning feedback with the corresponding interaction states. Together, these results support learning from coherent state transitions and aligned feedback, rather than treating trajectories as unordered collections of observations.

Candidate Quality under Fixed Proposal Budgets. To examine whether our method improves candidate selection beyond simply retaining more proposals, we compare True Target@K and Query-Matching Share@K under fixed proposal budgets. The former measures the fraction of queries whose top-K proposals contain the intended target, while the latter measures the proportion of selected proposals matching the query. As shown in Figure 3, the three methods perform similarly at $K = 1$. For $1 < K < 5$, ordered trajectory training achieves a faster increase in True Target@K and a smaller decline in Query-Matching Share@K than Direct and Shuffled pairs. These trends demonstrate the effectiveness of our method in retaining plausible candidates, improving target coverage while limiting the inclusion of non-matching candidates. For $K > 5$, Query-Matching Share@K declines more sharply, and the differences between methods narrow. This pattern is consistent with the limited number of query-matching objects in each scene. As the candidate number exceeds this number, additional candidates necessarily include non-matching objects. These results suggest that trajectory supervision improves the composition of the initial candidate set rather than obtaining higher coverage solely by retaining more proposals, providing a more reliable starting point for subsequent clarification.

5 Conclusion

In this paper, we formulate interactive 3D visual grounding as sequential candidate refinement and propose trajectory-level learning to retain plausible candidates until sufficient disambiguating evidence becomes available. To effectively propagate trajectory-level supervisory signals to early candidate generation, we introduce a trajectory-aware adapter that refines proposal representations and grounding scores using interaction history and candidate belief. We evaluate our framework on AmbiRefer3D and construct a more challenging subset to assess candidate preservation and multi-round disambiguation. Experiments across four grounding backbones demonstrate improved target retention and final grounding accuracy, highlighting the importance of learning how to avoid prematurely discarding candidates for reliable interactive 3D visual grounding. We believe our efforts will facilitate future research on trajectory-aware grounding and embodied interaction under incomplete information.

AI USE STATEMENT

We used generative AI tools, including Codex, to assist with developing and refining the conceptual framework, providing feedback on experimental designs, and implementing and debugging code. We also used these tools for literature discovery, manuscript drafting and revision, translation, LaTeX formatting, and figure preparation, as well as for checking the manuscript for clarity, consistency, and potential errors. All AI-assisted outputs and suggested revisions were reviewed and verified by the authors. We take responsibility for the final manuscript, experimental claims, code, and other research artifacts.

REPRODUCIBILITY STATEMENT

The main paper describes the framework, training objective, metric definitions and evaluation protocol. Appendix A supplements the adapter architecture and candidate-state updates, while Appendix A.2 describes constraint selection, candidate filtering, and stopping. Loss construction, and preference comparisons are detailed in Appendix A.3. Appendix C documents the construction of the challenging subset, and Appendix B specifies the training schemes and ablation settings.

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 *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 422–440, 2020. URL https://www.ecva.net/papers/eccv_2020/papers_ECCV/papers/123460409.pdf.

Dave Zhenyu Chen, Angel X. Chang, and Matthias Nießner. ScanRefer: 3D object localization in RGB-D scans using natural language. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 202–221, 2020. doi: 10.1007/978-3-030-58565-5_13. URL https://arxiv.org/abs/1912.08830.

Shizhe Chen, Pierre-Louis Guhur, Cordelia Schmid, and Ivan Laptev. History aware multimodal transformer for vision-and-language navigation. *Advances in neural information processing systems*, 34:5834–5847, 2021.

Xiao Chen, Quanyi Li, Tai Wang, Tianfan Xue, and Jiangmiao Pang. Gennbv: Generalizable next-best-view policy for active 3d reconstruction. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16436–16445. IEEE, 2024.

Duc Cao Dinh, Khai Le-Duc, Florent Draye, Chris Ngo, Terry Jingchen Zhang, Bernhard Schölkopf, and Zhijing Jin. Pruneground: Plug-and-play spatial pruning for 3d visual grounding. *arXiv preprint arXiv:2606.31148*, 2026.

Hongjie Gu, Jinlong Fan, Liang Zheng, Jing Zhang, and Yuxiang Yang. Ddpa-3dvg: vision-language dual-decoupling and progressive alignment for 3d visual grounding. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence*, pp. 1044–1052, 2025.

Wenxuan Guo, Xiuwei Xu, Ziwei Wang, Jianjiang Feng, Jie Zhou, and Jiwen Lu. Tsp3d: text-guided sparse voxel pruning for efficient 3d visual grounding. *arXiv preprint arXiv:2502.10392*, 2025.

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 with GPT and prototype guidance. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 15372–15383, 2023. URL https://openaccess.thecvf.com/content/ICCV2023/html/Guo_ViewRefer_Grasp_the_Multi-view_Knowledge_for_3D_Visual_Grounding_ICCV_2023_paper.html.

Yue Hu, Avery Xi, Qixin Xiao, Seth Isaacson, Henry X Liu, Ram Vasudevan, and Maani Ghaffari. Longnav-r1: Horizon-adaptive multi-turn rl for long-horizon vla navigation. *arXiv preprint arXiv:2602.12351*, 2026.

Haifeng Huang, Xinyi Chen, Yilun Chen, Hao Li, Xiaoshen Han, Zehan Wang, Tai Wang, Jiangmiao Pang, and Zhou Zhao. RoboGround: Robotic manipulation with grounded vision-language priors. *arXiv preprint arXiv:2504.21530*, 2025. URL https://arxiv.org/abs/2504.21530.

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

Dinesh Jayaraman and Kristen Grauman. Learning to look around: Intelligently exploring unseen environments for unknown tasks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1238–1247. IEEE, 2018.

Rithvik Jonna, Aakash Gurram, Man Namgung, Wyatt Mackey, and Tinoosh Mohsenin. Bit-nav: Brain-inspired trajectory memory for embodied navigation. *arXiv preprint arXiv:2606.21398*, 2026.

Taichi Kato, Takuya Kiyokawa, Namiko Saito, and Kensuke Harada. Replanning human-robot collaborative tasks with vision-language models via semantic and physical dual-correction. *arXiv preprint arXiv:2602.14551*, 2026. URL https://arxiv.org/abs/2602.14551.

Haijie Li, Jiaxin Zhang, Dave Zhenyu Chen, Youyu Chen, Yanmin Wu, and Jian Zhang. Coordrefer: Coordinate-aware 3d visual grounding from multiview images. *arXiv preprint arXiv:2608.05569*, 2026.

Rong Li, Shijie Li, Lingdong Kong, Xulei Yang, and Junwei Liang. Seeground: See and ground for zero-shot open-vocabulary 3d visual grounding. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3707–3717. IEEE, 2025.

Tianyi Lu, Hui Zhang, Zijie Diao, Junke Wang, Shengqi Xu, Xingyao Lin, Guojin Zhong, Ziyi Ye, Peng Wang, Zuxuan Wu, and Yu-Gang Jiang. ThinkingVLA: Interleaved vision and language reasoning for robotic manipulation. *arXiv preprint arXiv:2606.17937*, 2026. URL https://arxiv.org/abs/2606.17937.

Ziyang Lu, Yunqiang Pei, Guoqing Wang, Peiwei Li, Yang Yang, Yinjie Lei, and Heng Tao Shen. ScanERU: Interactive 3d visual grounding based on embodied reference understanding. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(4):3936–3944, 2024. doi: 10.1609/aaai.v38i4.28186. URL https://ojs.aaai.org/index.php/AAAI/article/view/28186.

Kun Luo, Xiangyu Dong, Xiaoguang Ma, Haoran Zhao, and Yaoming Zhou. LightZeroNav: Zero-shot vision language navigation in continuous environments based on lightweight VLMs. *arXiv preprint arXiv:2603.16947*, 2026. URL https://arxiv.org/abs/2603.16947.

Zhangyang Qi, Zhixiong Zhang, Yizhou Yu, Jiaqi Wang, and Hengshuang Zhao. Vln-r1: Vision-language navigation via reinforcement fine-tuning. *arXiv preprint arXiv:2506.17221*, 2025.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741, 2023.

Mohit Shridhar and David Hsu. Interactive visual grounding of referring expressions for human-robot interaction. In *Proceedings of Robotics: Science and Systems*, Pittsburgh, Pennsylvania, June 2018. doi: 10.15607/RSS.2018.XIV.028. URL https://roboticsproceedings.org/rss14/p28.html.

Xuefei Sun, Xujia Zhang, Brendan Crowe, Doncey Albin, and Christoffer Heckman. Scenegraph-grounder: Zero-shot 3d visual grounding via structured scene graph matching. *arXiv preprint arXiv:2605.21788*, 2026.

Jun Wang, Xiaohao Xu, and Xiaonan Huang. Probing collision grounding in vision-language models for safe human-robot collaboration. *arXiv preprint arXiv:2605.31196*, 2026a. URL https://arxiv.org/abs/2605.31196.

Shuo Wang, Yucheng Wang, Guoxin Lian, Yongcai Wang, Maiyue Chen, Kaihui Wang, Bo Zhang, Zhizhong Su, Yutian Zhou, Wanting Li, et al. Progress-think: Semantic progress reasoning for vision-language navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4076–4086, 2026b.

Xinyi Wang, Na Zhao, Zhiyuan Han, Dan Guo, and Xun Yang. Augrefer: Advancing 3d visual grounding via cross-modal augmentation and spatial relation-based referring. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 8006–8014, 2025.

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, 2023. URL https://openaccess.thecvf.com/content/CVPR2023/html/Wu_EDA_Explicit_Text-Decoupling_and_Dense_Alignment_for_3D_Visual_Grounding_CVPR_2023_paper.html.

Can Xu, Yuehui Han, Rui Xu, Le Hui, Jin Xie, and Jian Yang. Multi-attribute interactions matter for 3d visual grounding. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 17253–17262. IEEE, 2024a.

Jie Xu, Hanbo Zhang, Qingyi Si, Yifeng Li, Xuguang Lan, and Tao Kong. Towards unified interactive visual grounding in the wild. In *2024 IEEE International Conference on Robotics and Automation*, pp. 3288–3295, 2024b. doi: 10.1109/ICRA57147.2024.10611354. URL https://arxiv.org/abs/2401.16699.

Kuan Xu, Ruimeng Liu, Yizhuo Yang, Denan Liang, Tongxing Jin, Shenghai Yuan, Chen Wang, and Lihua Xie. A deployable embodied vision-language navigation system with hierarchical cognition and context-aware exploration. *arXiv preprint arXiv:2604.21363*, 2026. URL https://arxiv.org/abs/2604.21363.

Li Yang, Ziqi Zhang, Zhongang Qi, Yan Xu, Wei Liu, Ying Shan, Bing Li, Weiping Yang, Peng Li, Yan Wang, et al. Exploiting contextual objects and relations for 3d visual grounding. *Advances in Neural Information Processing Systems*, 36:49542–49554, 2023.

Hanbo Zhang, Jie Xu, Yuchen Mo, and Tao Kong. InViG: Benchmarking open-ended interactive visual grounding with 500k dialogues. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 5508–5518, June 2024. URL https://openaccess.thecvf.com/content/CVPR2024W/JRDB/html/Zhang_InViG_Benchmarking_Open-Ended_Interactive_Visual_Grounding_with_500K_Dialogues_CVPRW_2024_paper.html.

Yiming Zhang, Zeming Gong, and Angel X. Chang. Multi3DRefer: Grounding text description to multiple 3D objects. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 15225–15236, 2023. URL https://openaccess.thecvf.com/content/ICCV2023/html/Zhang_Multi3DRefer_Grounding_Text_Description_to_Multiple_3D_Objects_ICCV_2023_paper.html.

Lichen Zhao, Daigang Cai, Lu Sheng, and Dong Xu. 3DVG-Transformer: Relation modeling for visual grounding on point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 2928–2937, 2021. URL https://openaccess.thecvf.com/content/ICCV2021/html/Zhao_3DVG-Transformer_Relation_Modeling_for_Visual_Grounding_on_Point_Clouds_ICCV_2021_paper.html.

Duo Zheng, Shijia Huang, and Liwei Wang. Video-3d llm: Learning position-aware video representation for 3d scene understanding. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8995–9006. IEEE, 2025.

Rongjiang Zhu, Wei Kang, Zeqi Liu, Junyu Chen, Shuo Yang, and Xinxiao Wu. AmbiRefer3D: 3d visual grounding with referential ambiguity. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://yearnallover.github.io/ambirefer/.

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 *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 2899–2909. IEEE, 2023.

A Additional Method Details

A.1 Candidate Generation and Adapter Implementation

Backbone interface. We refer to the object proposals produced by the grounding backbone as candidates. Backbone-specific interfaces expose candidate boxes, representations, grounding logits, and query embeddings in a common format. Available cross-modal features $\mathbf{Z}^t$ are concatenated with adapter-refined representations before Interactive Disambiguation,

Candidate distributions. To combine previous filtering with current query matching, we weight sigmoid grounding scores by incoming retention and normalize across valid candidates. Define $\mathcal{N}_m(\mathbf{x})_i = m_i x_i / \max(\sum_j m_j x_j, \epsilon)$ for valid weights, where $\epsilon = 10^{-6}$ prevents division by zero. The distributions before and after score refinement are

$$\begin{aligned} \boldsymbol{\omega}^t_{\text{before}} &= \mathcal{N}_m\big(\boldsymbol{\alpha}^{t-1}\sigma(\boldsymbol{\ell}^t)\big)\,, \\ \boldsymbol{\omega}^t &= \mathcal{N}_m\Big(\boldsymbol{\alpha}^{t-1}\sigma(\widetilde{\boldsymbol{\ell}}^t)\Big)\,. \end{aligned} \tag{10}$$

Both use the same retention weights; only the grounding scores change. These are relative candidate weights, not calibrated target probabilities.

Feature aggregation and geometric context. The pre-refinement distribution determines which candidates contribute to the history summary and spatial context. For box centers $\mathbf{c}_i$ and positive side lengths $\mathbf{s}_i$, let $\boldsymbol{\delta}_{ij} = \mathbf{c}_j - \mathbf{c}_i$ and $r_{ij} = \max(\|\boldsymbol{\delta}_{ij}\|_2, 10^{-4})$. We represent each pair by

$$\boldsymbol{\xi}_{ij} = [\boldsymbol{\delta}_{ij}; r_{ij}; \boldsymbol{\delta}_{ij}/r_{ij}; \log \mathbf{s}_j - \log \mathbf{s}_i]. \tag{11}$$

This descriptor encodes relative position, distance, direction, and log-size ratios. A learned geometric encoder φ and candidate projection P_v give

$$\begin{aligned} \overline{\mathbf{v}}^t &= \sum_i \omega^t_{\text{before},i} P_v(\mathbf{v}_i), \\ \mathbf{r}^t_i &= \sum_j \bar{\omega}^t_{ij} \varphi(\boldsymbol{\xi}_{ij}). \end{aligned} \tag{12}$$

Here, the geometric encoder φ is a two-layer MLP with 64 hidden units and a 128-dimensional output. The projection P_v maps candidate representations to the same 128-dimensional feature space. $\bar{\omega}^t_{ij}$ is $\omega^t_{\text{before},j}$ normalized over valid neighbors with the same numerical safeguard. The shared feature summary enters the history update, whereas each $\mathbf{r}^t_i$ enters the correction heads for candidate i.

Filtering statistics and history. To track how the candidate distribution evolves, we measure its concentration and change from the previous refined distribution:

$$\begin{aligned} H^t &= -\sum_i \omega^t_{\text{before},i} \log \max(\omega^t_{\text{before},i}, \epsilon), \\ D^t &= \sum_i |\omega^t_{\text{before},i} - \omega^{t-1}_i|. \end{aligned} \tag{13}$$

Normalized distributions alone cannot show whether all retention weights have decreased. We therefore also compute

$$R^t = \frac{\sum_i m_i \alpha^{t-1}_i}{\max(\sum_i m_i, 1)}. \tag{14}$$

R^t averages soft weights rather than counting retained objects. The statistics and feedback form $\mathbf{f}^t = [u^t; H^t; D^t; R^t]$. The history update combines their learned embedding with the query, candidate summary, and previous history:

$$\mathbf{h}^t = \mathrm{GRUCell}\left([P_q(\mathbf{e}^t); \overline{\mathbf{v}}^t; P_f(\mathbf{f}^t); \mathbf{h}^{t-1}],\ \mathbf{h}^{t-1}\right). \tag{15}$$

Here, P_q consists of layer normalization, a linear projection, and GELU, mapping the query embedding to 128 dimensions. P_f is a linear projection followed by GELU that maps the feedback sign and three filtering statistics to a 128-dimensional feature vector. The GRU maintains a 128-dimensional history state. Semicolons indicate concatenation. The previous history is included both in the concatenated input and as the recurrent state. P_q projects the query and P_f encodes the four feedback and filtering statistics. Initially, $\boldsymbol{\alpha}^{-1} = \mathbf{m}$, $\mathbf{h}^{-1} = \mathbf{0}$, $u^0 = 0$, and $\boldsymbol{\omega}^{-1} = \boldsymbol{\omega}^0_{\mathrm{before}}$, giving $D^0 = 0$.

Feedback-conditioned corrections. Two heads receive $[P_v(\mathbf{v}_i); \mathbf{r}_i^t; \mathbf{h}^t]$ and predict bounded feature and score corrections for the residual update. To limit score suppression before affirmative clarification, the negative-correction gate is

$$g^t = g_{\min} + (1 - g_{\min})\,\mathrm{clip}(u^t, 0, 1)\sigma(G_f(\mathbf{f}^t)). \tag{16}$$

G_f is a learned gate predictor and $g_{\min} = 0.05$. The gate is shared across candidates. Absent or negative feedback leaves it at its minimum, while preferred feedback can increase it.

After score refinement, the second normalization in Eq. 10 produces $\boldsymbol{\omega}^t$. The retention update in the main text follows during training; initial hard filtering instead constructs $\mathcal{C}^0$ for interaction.

A.2 Interactive Disambiguation

State and prediction heads. The state combines the query, candidate features and geometry, grounding scores, retention weights, candidate distribution, and constraint context. Other scene candidates can provide landmark context without becoming eligible predictions. The retention head attends from candidate features to the query and selected constraint; the stopping head combines the state, query, constraint, and filtering statistics.

Feedback and rollout. Training processes cumulative queries and carries adapter history and retention across rounds. The interaction branch uses initial-query features and scores with trajectory states. The training mixes supplied and predicted states and includes annotated matching candidates in the training decision set.

Filtering and stopping. Constraint selection is masked to available, unused actions. The adaptive threshold is compared with the current retention score $\sigma(k_i^t)$, not the cumulative score. A safeguard keeps at least one candidate and, unless a predicted stop is permitted, at least half of the current set. Previously discarded candidates cannot re-enter.

The initial distribution $\mathbf{p}^0$ is normalized over $\mathcal{C}^0$. Cumulative scores start from $\log p_i^0$ and add per-round log-retention scores, determining the final ranking among survivors. Stopping is checked after candidate filtering using the predicted stop probability, remaining candidates, and available constraints.

A.3 Training Losses and Preferences

We jointly optimize candidate generation and interactive disambiguation with

$$\mathcal{L} = \mathcal{L}_{\mathrm{cg}} + \lambda_{\mathrm{ret}}\mathcal{L}_{\mathrm{ret}} + \lambda_{\mathrm{int}}\left(\mathcal{L}_{\mathrm{sup}} + \lambda_{\mathrm{pref}}\mathcal{L}_{\mathrm{ipref}}\right). \tag{17}$$

The four terms supervise grounding, trajectory retention, interaction decisions, and trajectory-derived preferences, respectively. We use $\lambda_{\mathrm{ret}} = 0.2$, $\lambda_{\mathrm{int}} = 1$, and $\lambda_{\mathrm{pref}} = 0.5$.

Grounding supervision. $\mathcal{L}_{\mathrm{cg}}$ retains each backbone's original objectives rather than imposing a shared detection loss on all four backbones.

Retention supervision. Let $\mathcal{P}_\star$ denote candidates assigned to the intended target using IoU 0.5, and $\mathcal{T}_v$ the valid observations. To penalize discarding all target-matched candidates, we use

$$\mathcal{L}_{\text{surv}} = -\frac{1}{|\mathcal{T}_v|} \sum_{t \in \mathcal{T}_v} \log \max \left(1 - \prod_{i \in \mathcal{P}_\star} (1 - \alpha_i^t), \epsilon \right). \tag{18}$$

This is a differentiable target-preservation surrogate. Trajectories without a target-matched candidate before filtering are excluded from this supervision.

The complete retention objective combines preservation with progressive refinement:

$$\begin{aligned} \mathcal{L}_{\text{ret}} = \mathcal{L}_{\text{surv}} + \mathcal{L}_{\text{rank}} + 0.1\mathcal{L}_{\text{set}} + 0.25\mathcal{L}_{\text{qpair}} \\ + 0.5\big(\mathcal{L}_{\text{prune}} + \mathcal{L}_{\text{transition}} + \mathcal{L}_{\text{improve}} + \mathcal{L}_{\text{KL}}\big). \end{aligned} \tag{19}$$

Here, $\mathcal{L}_{\text{prune}}$ reduces final retention weights of distractors, and $\mathcal{L}_{\text{rank}}$ ranks the target above distractors at the final observation. $\mathcal{L}_{\text{set}}$ applies class-balanced binary supervision to raw logits shifted by $\text{logit}(\eta_1)$, protecting all candidates matching the accumulated query. $\mathcal{L}_{\text{transition}}$ supervises feedback-consistent changes in target score mass, while $\mathcal{L}_{\text{improve}}$ encourages its increase from the initial to the final observation. These terms use a softplus margin of 0.2 and stop gradients through earlier comparison scores. $\mathcal{L}_{\text{qpair}}$ uses a reference-free logistic comparison of summed target log masses for aligned original and constructed negative-feedback query sequences, with scale 0.1. Finally, $\mathcal{L}_{\text{KL}}$ regularizes refined initial sigmoid scores toward detached raw scores using Bernoulli KL divergence. Intermediate matching sets may contain multiple objects.

Supervised interaction. $\mathcal{L}_{\text{sup}}$ combines constraint-selection, candidate-filtering, and stopping losses with weights 1, 1, and 0.2, respectively. Selection uses cross-entropy with recorded actions or utility-derived soft targets at temperature 0.1. Filtering uses binary supervision on retention logits shifted by $\text{logit}(\eta_2)$, with separately averaged positive and negative terms weighted 3 and 0.25. An additional negative log matching-set mass term has weight 0.5. Stopping uses binary cross-entropy with annotated stop/continue labels. Local ranking and soft metric auxiliaries each have weight 1.

Preference construction. We compare observed alternative actions from the same scene, expression, intended target, and annotated candidate state. For a record with J actions, action j receives utility

$$U_j = V_j - c(J - j - 1) + \lambda_I \sum_{r=j}^{J-1} \gamma^{r-j} d_r. \tag{20}$$

Here, d_r is the fractional decrease in the annotated candidate count at step r, clipped to $[0, 1]$. $V_j = -1$ if any subsequent annotated set loses the target; otherwise, it is γ^{J-j-1} times terminal target presence. We use $\gamma = 0.9$, $c = 0.1$, and $\lambda_I = 0.25$. This utility rewards target preservation and candidate reduction while penalizing additional rounds. Its terminal signal is annotated target presence, not measured model accuracy.

Utilities are averaged over repeated state–action observations. An observed action is preferred to another at the same state when its utility is higher by at least 0.01. States without eligible alternatives contribute no action pair. We also compare continued clarification with premature stopping at initial states annotated as requiring further interaction.

DPO-inspired preference optimization. For logits $\mathbf{s}$, define the log score mass assigned to a matching set $\mathcal{P}$ within decision set $\mathcal{D}$:

$$\mathcal{M}(\mathbf{s}, \mathcal{P}; \mathcal{D}) = \log \frac{\sum_{i \in \mathcal{P} \cap \mathcal{D}} \exp(s_i)}{\sum_{i \in \mathcal{D}} \exp(s_i)}. \tag{21}$$

The composite comparison score combines selection, filtering, grounding, and stopping:

$$\begin{aligned} S_\Theta(\tau) = \sum_t \delta_t \big[\log \pi_\psi(a^t \mid s^t) + \mathcal{M}(\mathbf{k}^t, \mathcal{P}_{t+1}; \mathcal{D}_t) \\ + \lambda_g \mathcal{M}(\boldsymbol{\ell}_{\text{in}}^t, \mathcal{P}_t; \mathcal{D}_t) + B_t \big]. \end{aligned} \tag{22}$$

Here, δ_t masks eligible comparison steps, $\mathcal{P}_t$ and $\mathcal{P}_{t+1}$ are the annotated matching sets before and after the action, and $\lambda_g = 1$. $\boldsymbol{\ell}^t_{\text{in}}$ contains the grounding logits supplied to the interaction module. B_t is the log probability of the annotated stop/continue label. Empty matching intersections contribute zero to their mass term.

Preferred and rejected comparisons τ^+ and τ^- are optimized with

$$\mathcal{L}_{\text{pair}} = -\log \sigma\Big(\beta\big[S_\Theta(\tau^+) - S_\Theta(\tau^-) - S_{\text{ref}}(\tau^+) + S_{\text{ref}}(\tau^-)\big]\Big). \tag{23}$$

We use $\beta = 0.1$, which scales the preference margin rather than weighting the loss in the overall objective. The reference is a frozen copy of Interactive Disambiguation evaluated without gradients on current grounding inputs. For continue-versus-stop comparisons, the scores are $\log \sigma(-z^0)$ and $\log \sigma(z^0)$, respectively, with the same reference correction. $\mathcal{L}_{\text{ipref}}$ averages the active comparison losses.

Rejected action inputs reuse the supplied state and annotations with an alternative constraint; they are not newly generated counterfactual rollouts. We therefore use DPO-inspired optimization of composite decision scores, rather than standard DPO on normalized dialogue likelihoods.

B Training Controls

B.1 Three Training Schemes

Direct. The pretrained grounding backbone is frozen and the adapter is disabled. Interactive Disambiguation uses independent rounds with selection, filtering, and stopping supervision. Trajectory retention and sequence-preference objectives are disabled. No interaction gradient updates Candidate Generation.

Shuffled pairs. Source trajectory rounds become independent examples, retaining their within-round query and matching-set labels. History, retention, and candidate distributions are reset before each observation. The adapter is enabled, and trainable grounding components can receive joint gradients. Local grounding, matching-set, ranking, and metric losses remain active; terminal suppression and ranking become per-observation objectives. Cross-round transition, improvement, sequence preferences, and the configured initial consistency term are disabled.

Ordered trajectory. Recorded observations are processed in order, carrying history, candidate distributions, and soft retention across rounds. The retention and preference objectives described above are enabled. The interaction branch uses initial grounding features with trajectory states and mixes supplied and predicted states equally during autoregressive training. Trainable grounding, adapter, and interaction parameters are optimized jointly.

B.2 Trajectory-Information Ablations

Shuffle round order. The initial grounding observation stays fixed, while subsequent observations are permuted as complete records, including aligned query, matching-set, and feedback fields. Trajectories with only one post-initial observation are unchanged. Interaction action rows are separately permuted and reindexed. Unlike Shuffled pairs, recurrent processing remains active, but the original transition order is disrupted.

Shuffle feedback. Non-initial signs are shuffled across original and constructed negative-feedback training slots. Global sign counts are preserved, while queries, matching sets, and interaction action rows remain unchanged. This breaks sign–observation alignment. Negative slots are essential: shuffling affirmative signs alone would leave the data unchanged. The control tests constructed feedback alignment, not unrestricted human-feedback robustness.

Both interventions modify training records only. Evaluation uses unchanged records, candidate thresholds, and the same interaction budget.

B.3 Component Ablations

Adapter. This switch enables recurrent feature and score corrections. Removing it bypasses residual refinement while preserving the grounding-to-interaction interface.

Retention supervision (Ret.). This switch controls survival, terminal suppression and ranking, raw matching-set, transition, and target-score improvement losses. Removing it sets these weights to zero.

Gate. Removing the feedback gate sets $g^t = 1$, leaving negative score corrections unattenuated. It does not remove the separate adaptive threshold or minimum-retention safeguard. The gate requires the adapter.

Preference supervision (DPO). The switch enables both the reference-relative interaction objective and the reference-free query-sequence comparison. Removing it disables both while preserving other supervision. Thus the column tests a preference package, not a single standard DPO objective.

C Dataset Construction

C.1 Challenging Multi-Round Subset

Construction objective and requirements. The subset evaluates target retention until sufficient clarification information becomes available. Construction starts with an ambiguous query and an intended target; scene edits introduce competing objects where needed. An accepted path satisfies four requirements: the initial query matches multiple objects; each nonterminal annotated set remains ambiguous; the target stays in nested matching sets; and at least two clarifications lead to a specific target. Clarifications cover object attributes, spatial relations, and landmark-relative rankings.

Subset size and human review. The subset contains 220 examples from 44 ScanNet validation scenes that cover the largest possible area. Each has its own scene instance: 195 include edits and 25 retain the source geometry. The 449 clarification annotations comprise 223 attribute and 226 relational actions. Excluding the initial query, 211 examples contain two clarifications and nine contain three. All 220 examples were manually reviewed for initial ambiguity, target identity, scene edits, clarification text, and candidate selections. Review checks target preservation, set nesting, intermediate ambiguity, and final uniqueness.

Visual annotation and review interface. The browser interface combines a 3D scene view, object assets, and a multi-round annotation panel. Annotators inspect the scene, select the target and matching candidates, and enter queries and successive clarifications. The interface supports adding, copying, moving, and deleting objects. Source-object identities and edit transforms are retained for reproducibility. Figure 4 illustrates the interface.

Annotation examples. Figures 5 and 6 show two- and three-round examples. Each shows the initial query, successive constraints, and annotated matching sets. These are dataset annotations, not model predictions.

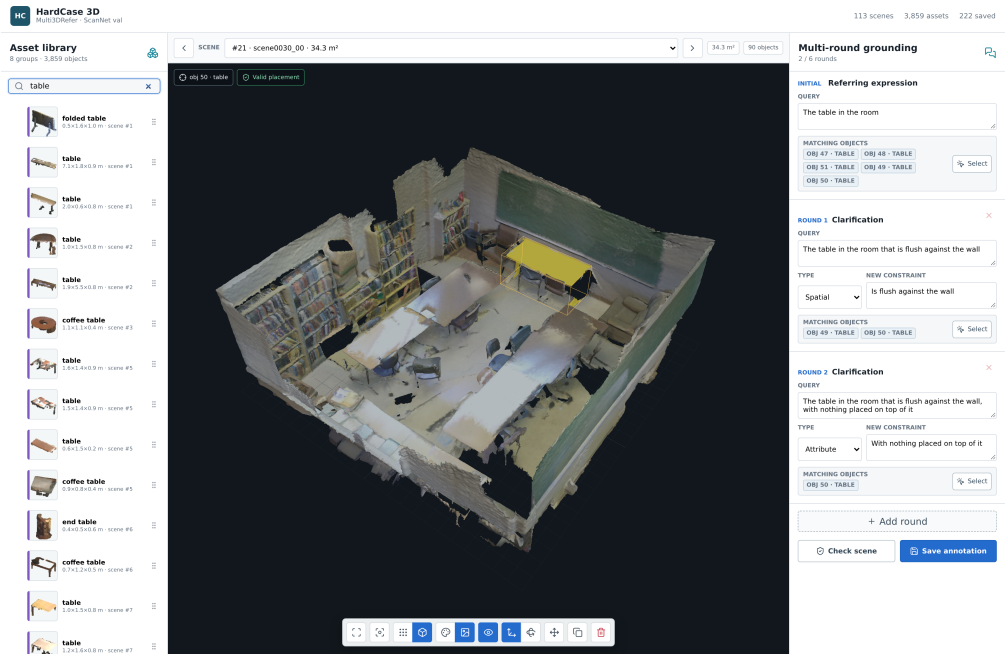


Figure 4: Scene-editing and multi-round candidate annotation interface.

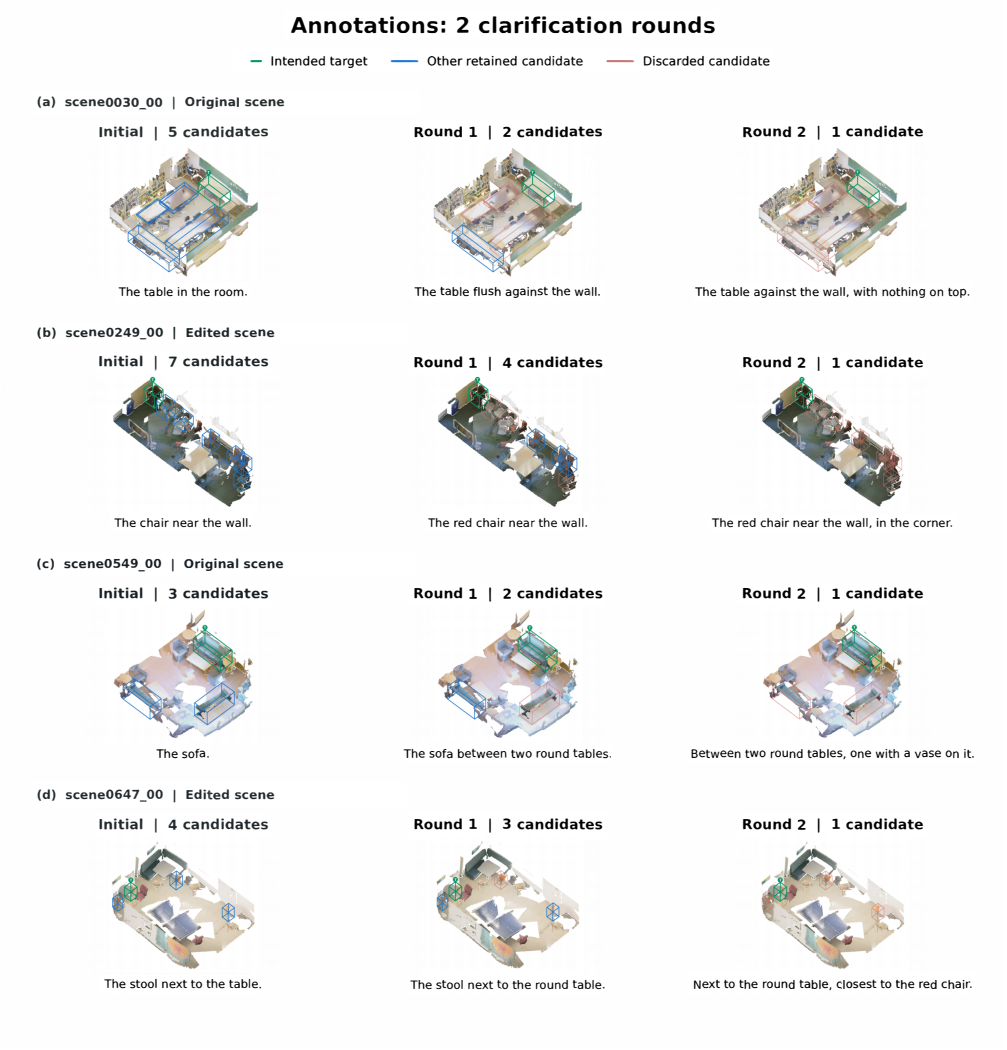


Figure 5: An annotated two-step path with ambiguity remaining after the first clarification.

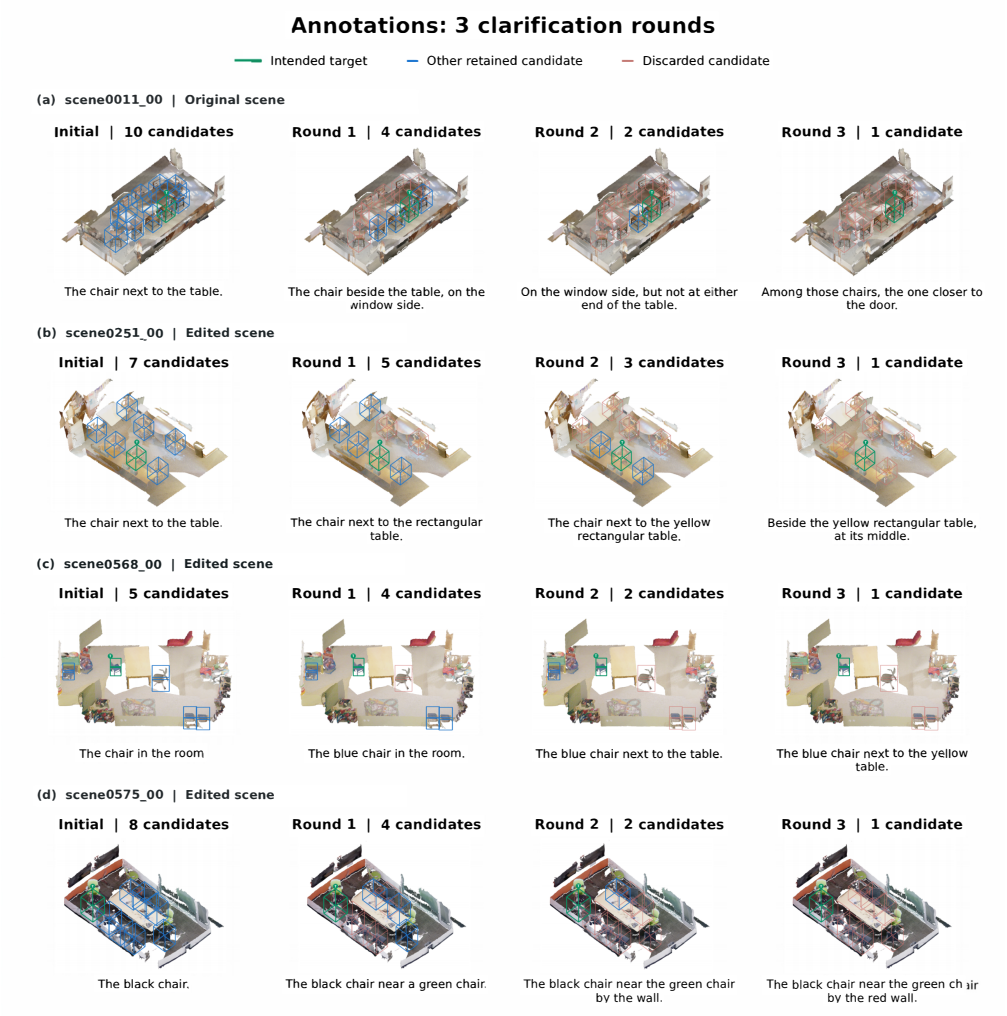


Figure 6: An annotated three-step path retaining the target until the final clarification.