ELSA: Execution-Feedback Reinforcement Learning for Spatial Language Parsing in 3D Visual Grounding

Anonymous ACL submission

Abstract

3D visual grounding (3DVG) is challenging due to the need to understand spatial language and 3D geometric relations among objects. While supervised approaches achieve strong performance, they require costly annotated 3D vision-language datasets. Training-free approaches based on LLMs eliminate annotation requirements, but rely on large language model APIs at every inference step, incurring substantial costs and preventing local deployment — while also failing to exploit the full potential of the geometric execution engine. To address these challenges, we introduce ELSA (Execution-feedback reinforcement Learning for Spatial lAnguage parsing), a framework that trains small language models (SLMs) via GRPO using execution outcomes as a natural, annotation-free reward signal, replacing LLM parsers with locally deployable SLMs that learn engine-specific spatial reasoning through grounding feedback. Experimental results demonstrate that ELSA achieves 69.3% accuracy on Nr3D, surpassing all prior training-free purely linguistic methods, at over 10× lower inference cost. Remarkably, our SLM surpasses the LLM-based baseline after only 350 RL steps (2,800 training samples), demonstrating that execution feedback enables SLMs to learn spatial relational reasoning with exceptional data efficiency. ELSA further generalizes zero-shot to ScanRefer and Sr3D without any additional fine-tuning.

1 Introduction

3D visual grounding (3DVG)—localizing objects in 3D scenes from natural language descriptions—is fundamental to embodied AI, yet it demands a nuanced understanding of spatial language across frames of reference, multi-hop relations, and observer-relative descriptions. While supervised approaches achieve strong performance (Zhu et al., 2023; Wu et al., 2023), they require large-scale annotated 3D vision-language data (Chen et al., 2020; Achlioptas et al., 2020). Training-free methods sidestep annotation costs by leveraging pretrained LLMs or VLMs, but introduce their own limitations: VLM-based methods (Xu et al., 2024; Li et al., 2025; Jin et al., 2026) require visual computation, while purely linguistic methods (Fang et al., 2024; Mi et al., 2025) operating on 3D coordinates and category labels alone, rely on large language model APIs at every inference step, incurring substantial cost and preventing local deployment.

Among purely linguistic methods, LaSP (Mi et al., 2025) achieves strong accuracy by parsing queries into symbolic programs executed by geometric encoders, but requires GPT-4o at inference. More fundamentally, zero-shot LLM parsing provides no feedback from execution outcomes: the parser never learns which spatial interpretations succeed with the engine, leaving its full potential unexploited.

We observe that the execution engine itself provides a direct, annotation-free training signal: whether the parsed program grounds to the correct object. As shown in Figure 1 (left), LaSP discards this signal entirely. ELSA instead routes execution outcomes back as reward, training small language models (SLMs) via GRPO (Shao et al., 2024) to replace LLM parsers with locally deployable models that learn engine-specific spatial reasoning. To address sparse rewards and signal absence during execution failures, we design a multi-level reward combining rank-based grounding feedback, semantic alignment, and format validity penalties.

Experiments show that, without any visual information, our 1.7B and 4B SLMs achieve 66.7% and 69.3% on Nr3D, surpassing LaSP with GPT-4o (65.7%) at over 10× lower inference cost (Figure 1, right). Both models transfer zero-shot to ScanRefer (63.9%) and Sr3D (97.8%), consistently outperforming prior methods across three benchmarks.

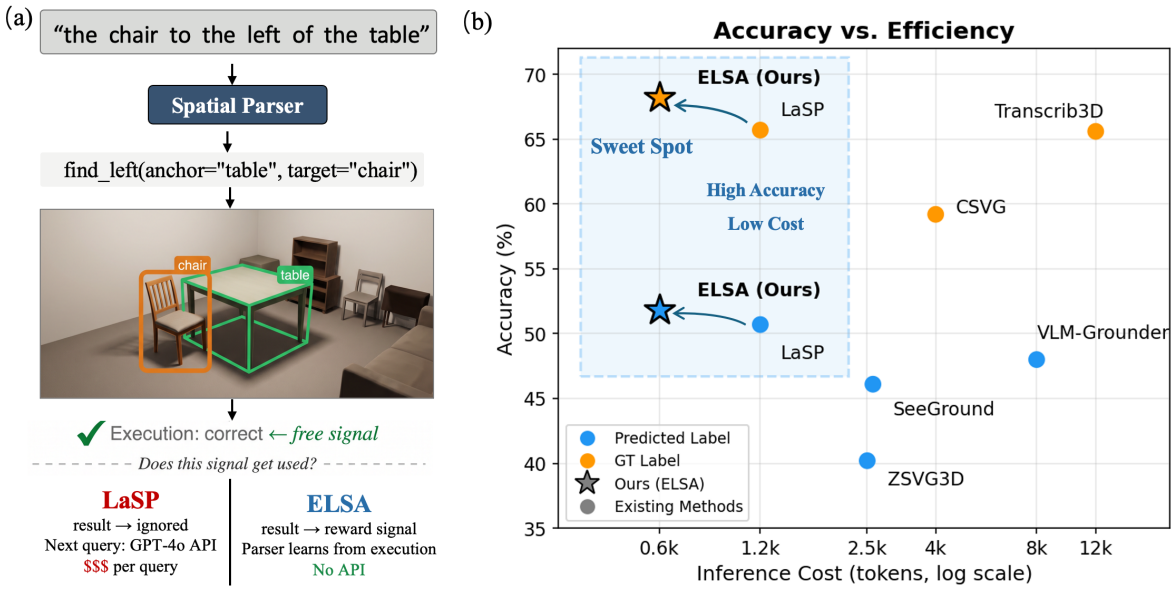

Figure 1: **(Left)** ELSA converts execution feedback into a reward signal. Given a natural-language query, a compact Spatial Parser generates a symbolic program; the program is executed on the 3D scene, and the outcome serves directly as a training reward, requiring no human annotation and no API calls. LaSP, by contrast, discards the execution result and invokes GPT-4o for every query. **(Right)** Accuracy vs. inference cost on Nr3D. ELSA (★) achieves high accuracy at a fraction of the inference cost of GPT-4o-based LaSP.

Remarkably, ELSA surpasses the LLM-based baseline after only 350 RL steps (≈2,800 samples): despite the complexity of spatial language, execution feedback provides a sufficiently rich signal for SLMs to learn spatial relational reasoning from very few examples.

Our contributions are:

- We propose **ELSA**, a framework training SLMs via execution-feedback GRPO for spatial language parsing, showing SLMs can surpass LLMs for this task at over 10× lower inference cost.
- We design a multi-level reward providing dense training signal for symbolic execution pipelines, addressing sparse rewards and execution failures.
- Without any visual model, our SLMs surpass LLM-based parsing on Nr3D and generalize zero-shot to ScanRefer and Sr3D, achieved with fewer than 3,000 training samples.

2 Related Work

2.1 Training-Free 3D Visual Grounding

Training-free 3DVG methods eliminate the need for annotated 3D vision-language data by leveraging pretrained LLMs or VLMs. Early work such as LLM-Grounder (Yang et al., 2024) decomposes referring expressions via LLMs and applies modular visual grounding. Transcrib3D (Fang et al., 2024) serialises 3D scenes as structured text and prompts GPT-4 to reason over object lists.

More recent methods incorporate richer visual information. VLM-Grounder (Xu et al., 2024) uses a VLM agent over rendered image sequences; SeeGround (Li et al., 2025) adapts 2D VLMs with perspective-aware rendering; SORT3D (Zantout et al., 2025) combines spatial reasoning toolboxes with sequential LLM inference; SPAZER (Jin et al., 2026) and Z3D (Drozdov et al., 2026) further leverage large VLMs with chain-of-thought reasoning. These methods achieve strong results but depend on visual models and repeated API calls.

LaSP (Mi et al., 2025) occupies a distinct position: it generates Python relation encoders offline via LLM and parses queries online with GPT-4o, achieving competitive accuracy through symbolic execution alone without any visual model. Our work builds directly on LaSP's framework, replacing its costly online LLM calls with an RL-trained SLM parser.

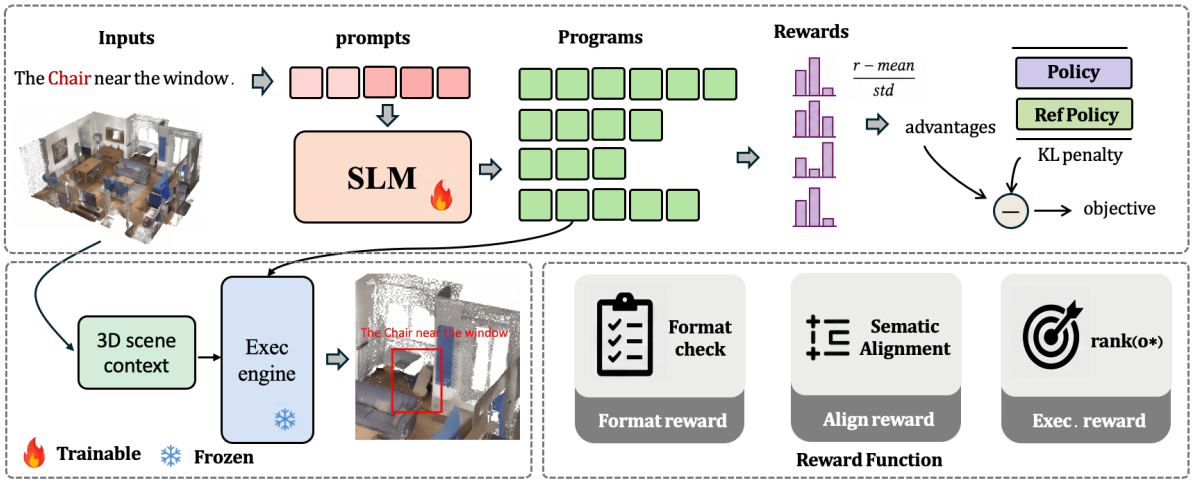

Figure 2: ELSA framework overview. **(Top)** Given a referring expression, the trainable SLM generates a group of G candidate symbolic programs from query input alone. Each program is passed to the frozen LaSP execution engine, which evaluates it against the 3D scene context; Execution outcomes are converted into rewards for GRPO training. **(Bottom right)** The reward function operates at three levels: format validity, semantic alignment, and rank-based execution accuracy.

2.2 Reinforcement Learning for Grounded Language

Reinforcement learning has been applied to grounded language tasks where execution or environment feedback provides training signal without intermediate annotation. RLHF (Ouyang et al., 2022) demonstrates that RL with outcome-based rewards can align language model behaviour with task objectives. In structured output generation, execution feedback has been used to train semantic parsers from question-answer pairs, bypassing the need for gold-standard logical forms (Liang et al., 2018).

Most directly related is Seg-Zero (Liu et al., 2025a), which applies GRPO to train a reasoning VLM for 2D image segmentation using IoU-based execution reward without annotated reasoning chains. ELSA extends this paradigm to 3D spatial language parsing: unlike Seg-Zero, our parser takes purely linguistic input with no visual encoding, and our execution engine is symbolic rather than neural.

2.3 Small Language Models via Reinforcement Learning

A growing body of work demonstrates that RL can train small language models (SLMs) to match or exceed much larger models on specific tasks. DeepSeek-R1 (Guo et al., 2025) shows that RL without chain-of-thought supervision elicits strong reasoning in smaller models. VisionReasoner (Liu et al., 2025b) extends this to multi-task visual perception. Our work contributes to this line by demonstrating that execution-feedback RL enables SLMs to surpass LLMs on spatial language parsing, a task requiring compositional understanding of 3D spatial relations.

3 Method

3.1 Overview

ELSA adopts a decoupled architecture: a trainable SLM that parses referring expressions into structured symbolic programs, and a frozen geometric execution engine that evaluates each program against the 3D scene. The SLM is trained via GRPO using grounding execution outcomes as the sole reward signal, with no intermediate annotation required (analogous to how Seg-Zero (Liu et al., 2025a) uses IoU from a frozen SAM model for 2D segmentation).

The SLM takes the referring expression as input and generates a symbolic program from text alone, without access to scene geometry. Scene-level information (object positions and categories) is provided exclusively to the execution engine, which converts the symbolic program into per-object scores using pre-generated Python geometric encoders. This separation makes the SLM responsible for language understanding and the engine responsible for geometric computation. Unlike continuous neural engines, the symbolic evaluator

can fail discretely: programs may be syntactically invalid, reference undefined relations, or specify categories absent from the scene. These failure modes motivate the multi-level reward design in §3.5. Figure 2 illustrates the full framework.

3.2 Preliminaries: Symbolic Execution for 3DVG

We build upon LaSP (Mi et al., 2025), a training-free 3DVG framework that decouples spatial language parsing from geometric computation. Given a 3D scene with N candidate objects (bounding box centres $\mathbf{c}_i \in \mathbb{R}^3$, category labels l_i) and a referring expression q, LaSP parses q into a hierarchical symbolic program P: a nested JSON specifying a target category and a list of spatial relation constraints. Each constraint names a spatial relation (e.g., `left`, `near`), a list of anchor objects in the same recursive format, and an optional negation flag, supporting multi-hop reasoning where anchors themselves carry spatial constraints.

The program is executed by relation encoders $\mathcal{E}_r$, which compute a score vector over all candidates. The predicted object is:

$$\hat{o} = \arg\max_i \ \mathrm{score}(i \mid P, \text{ scene}) \tag{1}$$

In LaSP's design, P is generated by GPT-4o at inference. Our goal is to train an SLM π_θ to produce P locally from the referring expression alone, replacing the API dependency with a compact locally deployable model.

3.3 Execution Outcome as a Training Signal

The key insight enabling our approach is that the execution engine provides a natural, annotation-free training signal: whether the parsed program grounds to the correct object o^*, already available in standard 3DVG datasets. Concretely:

$$r = f\big(\mathcal{E}(P, \text{ scene}),\ o^*\big) \tag{2}$$

where $P = \pi_\theta(q)$ is the SLM's parse. This requires no knowledge of why a parse succeeded or failed, no logical form supervision, and no additional annotation beyond what existing benchmarks provide. The signal is verifiable, scales to any annotated 3D scene, and directly measures the quantity we optimise, making it well-suited for reinforcement learning over spatial language parsing.

3.4 Training with GRPO

We optimise π_θ using Group Relative Policy Optimization (GRPO; Shao et al. 2024). For each referring expression q, we sample a group of G candidate programs $\{P_1, \ldots, P_G\}$ from the current policy. Each candidate receives a reward r_i from the execution engine. The within-group advantage for sample i is:

$$\hat{A}_i = \frac{r_i - \mathrm{mean}\big(\{r_j\}_{j=1}^G\big)}{\mathrm{std}\big(\{r_j\}_{j=1}^G\big)} \tag{3}$$

The policy is updated to increase the probability of high-advantage samples:

$$\mathcal{L}_{\mathrm{GRPO}}(\theta) = -\mathbb{E}\Big[\hat{A}_i \cdot \log \pi_\theta(P_i \mid q)\Big] + \beta \cdot D_{\mathrm{KL}}(\pi_\theta \| \pi_{\mathrm{ref}}) \tag{4}$$

GRPO requires no separate value network, reducing memory overhead compared to PPO. This is particularly advantageous when training on tens of thousands of 3DVG examples with structured JSON outputs.

3.5 Multi-Level Reward Design

Symbolic execution introduces failure modes absent in continuous neural engines: a generated program may be syntactically invalid, reference undefined relation types, or fail to execute when the predicted category is absent from the scene. A single binary reward provides no gradient signal in these cases. We address this with a **three-level reward** providing informative feedback at each stage of the parsing pipeline.

Level 1: Format Validity. The first level verifies that the output is a well-formed, executable symbolic program (valid JSON with at least one relation of correct arity). Programs that fail format checks receive tiered negative rewards and are not forwarded to the engine, enforcing structural validity from the first training step.

Level 2: Semantic Alignment. Before executing the program, we compute a lightweight alignment signal between the SLM's predicted relations and spatial keywords present in q:

$$r_{\text{align}} = \frac{|\mathcal{R}_{\text{pred}} \cap \mathcal{R}_{\text{expected}}|}{|\mathcal{R}_{\text{expected}}|} \tag{5}$$

where $\mathcal{R}_{\text{expected}}$ is extracted from q via keyword matching (e.g., "next to" $\rightarrow$ {`close`, `beside`}). When $\mathcal{R}_{\text{expected}} = \emptyset$, we set $r_{\text{align}} = 0.5$. This signal remains informative even when engine execution fails, providing a weak but non-zero gradient that encourages semantically appropriate relation choices throughout training.

Table 1: Nr3D grounding accuracy (%). Traditional methods use task-specific supervised 3D training (GT-label only). † uses rendered scene images. ∗ GT-label results reported by Zantout et al. (2025). **Bold**: best in each section.

Method	Overall	Easy	Hard	View Dep.	View Indep.
Traditional					
SAT (Yang et al., 2021)	49.2	56.3	42.4	46.9	50.4
BUTD-DETR (Jain et al., 2022)	54.6	60.7	48.4	46.0	58.0
3D-VisTA (Zhu et al., 2023)	64.2	72.1	56.7	61.5	65.1
ViL3DRef (Chen et al., 2022)	64.4	70.2	57.4	62.0	64.5
CoT3DRef (Abdelrahman et al., 2024)	64.4	70.0	59.2	61.9	65.7
Foundation Models (Predicted Label)					
SeeGround† (Li et al., 2025)	46.1	54.5	38.3	42.3	48.2
VLM-Grounder† (Xu et al., 2024)	48.0	55.2	39.5	45.8	49.4
ZSVG3D (Yuan et al., 2024b)	40.2	49.1	31.1	37.8	41.6
LaSP (w/o VLM) (Mi et al., 2025)	50.7	58.7	43.0	45.6	53.2
ELSA-1.7B (Ours)	50.2	58.1	42.5	46.7	51.9
ELSA-4B (Ours)	**51.8**	**60.4**	**43.6**	**47.9**	**53.8**
Foundation Models (Ground-Truth Label)					
CSVG (Yuan et al., 2024a)	59.2	59.2	44.5	53.0	46.4
Transcrib3D* (Fang et al., 2024)	65.6	—	—	63.3	66.7
LaSP (w/o VLM) (Mi et al., 2025)	65.7	75.6	56.2	58.7	69.1
ELSA-1.7B (Ours)	66.7	77.4	56.4	60.4	69.8
ELSA-4B (Ours)	**69.3**	**80.6**	**58.5**	**63.5**	**71.9**

Level 3: Execution Reward. Valid programs are executed by the LaSP engine. Rather than a binary signal, we use the rank of the ground-truth object o^* in the engine's output distribution:

$$r_{\text{exec}} = \begin{cases} 1 \,/\, \text{rank}(o^*) & \text{if rank}(o^*) \leq 5 \\ 0.2 & \text{otherwise} \end{cases} \tag{6}$$

This rank-based formulation provides continuous gradient signal: a parse ranking o^* second receives more reward than one ranking it fifth, incentivising progressive improvement.

Final Reward.

$$r = r_{\text{exec}} + \lambda \cdot r_{\text{align}} \tag{7}$$

The small weight λ ensures alignment acts as a regulariser. When execution fails entirely, the reward reduces to $0.1 \cdot r_{\text{align}}$, preserving a non-zero learning signal.

4 Experiment

4.1 Experimental Settings

Datasets. We conduct primary experiments on Nr3D (Achlioptas et al., 2020), which contains 41,503 human-annotated referring expressions grounded in ScanNet indoor scenes. Samples are stratified by distractor count into *easy* and *hard* subsets, and by linguistic perspective into *view-dependent* (VD) and *view-independent* (VI) subsets. We report top-1 accuracy under two conditions: *predicted-label*, where object categories are supplied by a 3D detector (ODIN), and *ground-truth-label*, where oracle categories are provided. The GT-label setting more directly measures spatial language parsing quality, which is the capability ELSA targets.

For cross-dataset generalization, we evaluate our Nr3D-trained models zero-shot on ScanRefer (Chen et al., 2020) and Sr3D (Achlioptas et al., 2020)—no data from either benchmark is used at any point during training.

Baselines. We compare ELSA against two groups. *Traditional* supervised methods train end-to-end on annotated 3D corpora: SAT (Yang et al., 2021), BUTD-DETR (Jain et al., 2022), 3D-VisTA (Zhu et al., 2023), ViL3DRef (Chen et al., 2022), and CoT3DRef (Abdelrahman et al., 2024). *Foundation model* methods require no task-specific 3D training. VLM-based methods (marked †) additionally use rendered scene images: SeeGround (Li et al., 2025) and VLM-Grounder (Xu et al., 2024). Text-only methods operate solely on object categories and 3D coordinates: ZSVG3D (Yuan et al., 2024b), CSVG (Yuan et al., 2024a), Transcrib3D (Fang et al., 2024), and LaSP (Mi et al., 2025). For LaSP, we use its VLM-free mode. Results for Transcrib3D under the GT-label setting are taken from the concurrent work SORT3D (Zantout et al., 2025), as the original paper does not report results under this evaluation protocol.

4.2 Main Results

Table 1 reports Nr3D accuracy under both evaluation settings.

Ground-truth-label. ELSA-4B achieves 69.3% overall, surpassing all baselines. Even ELSA-1.7B (66.7%) outperforms LaSP (65.7%), demonstrating that performance gains do not depend on model scale. Both SLMs improve consistently across Easy and Hard subsets and lead on View-Independent descriptions (ELSA-4B: 71.9%). On the View-Dependent subset, ELSA-4B (63.5%) improves over LaSP (58.7%) by 4.8 points, suggesting that execution feedback helps SLMs learn more viewpoint-aware spatial interpretations than zero-shot LLM parsing.

Predicted-label. Under the predicted-label setting, ELSA-4B (51.8%) leads all methods, improving over LaSP (50.7%) across all subsets. ELSA-1.7B (50.2%) remains competitive, with a notable gain on View-Dependent descriptions (46.7% vs. 45.6%). The consistent $\sim$17-point gap between GT-label and predicted-label accuracy confirms that detector error is orthogonal to spatial parsing quality, which is precisely the capability ELSA targets.

Qualitative Analysis. Figure 3 shows two representative cases where ELSA resolves queries that mislead zero-shot LLM parsing. In the first example, LaSP targets `box` from "*box of blankets*", while ELSA correctly identifies `blanket` as the referent. In the second, LaSP targets `magazine` rather than the intended `couch`, whereas ELSA correctly treats the magazines as spatial anchors. Both cases suggest that execution feedback trains SLMs to resolve referential intent rather than surface-level noun patterns.

Inference cost. Table 2 shows that ELSA runs at 0.06s per query, which is 35$\times$ faster than LaSP (2.1s) and over 400$\times$ faster than agent-based methods while consuming only 0.6k input tokens, half that of LaSP. Removing the API dependency also enables offline deployment and inference-time data privacy.

4.3 Cross-Dataset Generalisation

ScanRefer. Without any ScanRefer training data, ELSA-4B achieves 63.9% Acc@0.5 (Table 3), outperforming LaSP (61.7%) and all other baselines. ScanRefer descriptions are longer and more compositional than Nr3D, confirming that the spatial programs learned by ELSA generalise beyond the training distribution.

Table 2: Grounding cost comparison on a randomly sampled Nr3D subset. † uses commercial VLM API calls at inference time. **Bold**: best result.

Method	Time (s)	Tokens
ZSVG3D (Yuan et al., 2024b)	2.4	2.5k
SeeGround (Li et al., 2025)	9.0	2.6k
CSVG (Yuan et al., 2024a)	4.0	4.0k
VLM-Grounder (Xu et al., 2024)	50.3	8.0k
Transcrib3D (Fang et al., 2024)	27.0	12.0k
LaSP (w/o VLM) (Mi et al., 2025)	2.1	1.2k
ELSA-4B (Ours)	**0.06**	**0.6k**

Table 3: ScanRefer validation accuracy (%, GT bounding boxes). † evaluated on a partial subset. $*$ results reported by Zantout et al. (2025). **Bold**: best result.

Method	Acc@0.5
Chat-Scene (Huang et al., 2024)	50.3
3DGraphLLM (Zemskova and Yudin, 2025)	55.8
3D-VisTA† (Zhu et al., 2023)	55.6
QuatRoPE (Zhou et al., 2026)	58.2
LaSP (Mi et al., 2025)	61.7
Transcrib3D†$*$ (Fang et al., 2024)	62.0
ELSA-4B (Ours)	**63.9**

Sr3D. ELSA-4B achieves 97.8% on Sr3D (Table 4), and ELSA-1.7B (97.2%) also surpasses Transcrib3D$*$ (96.5%), demonstrating that the spatial programs learned on Nr3D transfer to Sr3D's allocentric referring expressions without any additional training.

4.4 Ablation Study

Effect of RL training. The zero-shot Qwen3-4B baseline achieves 58.00% without any RL training, while GRPO training lifts this to 69.31% (**+11.31 points**). This large improvement confirms that execution feedback is a highly effective training signal for spatial language parsing, enabling SLMs to substantially surpass both their own zero-shot performance and the GPT-4o-based LaSP baseline (65.7%).

Reward design. Replacing our rank-based reward with a binary 0/1 signal degrades performance by 0.79 points (68.52%), confirming that continuous gradient signal benefits training, particularly for queries where the correct object ranks second or third. Removing the alignment bonus causes a similar 0.36-point drop (68.95%), validating its role as a regulariser when execution feedback is

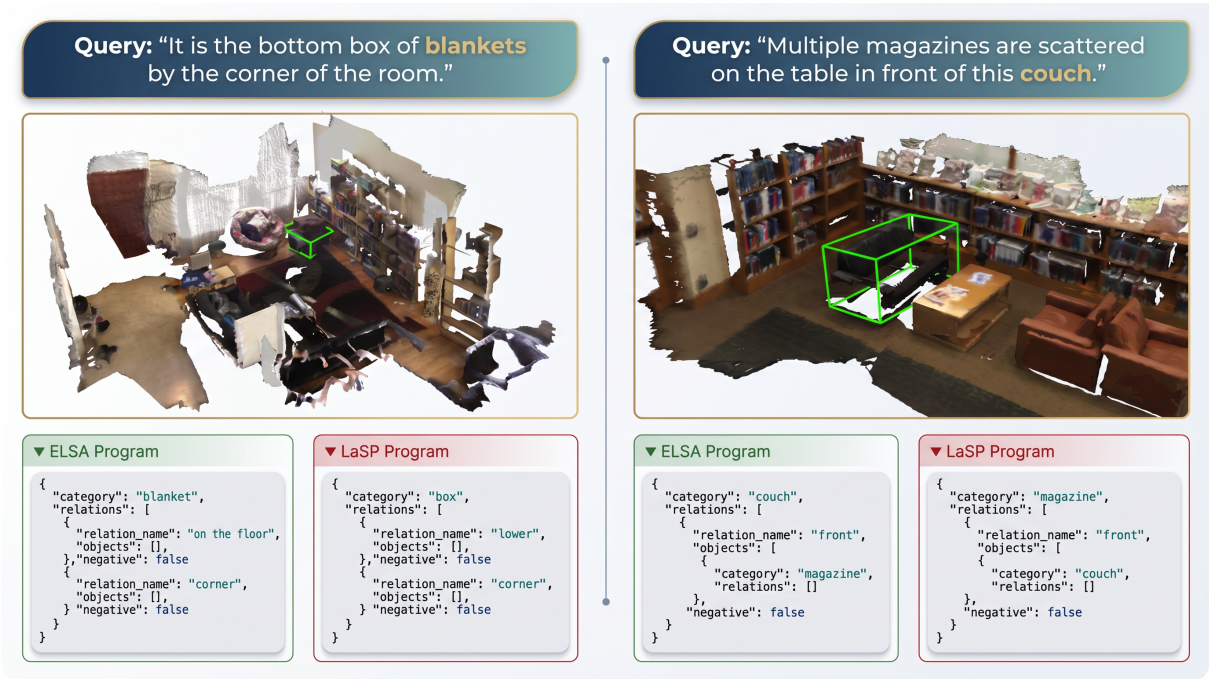

Figure 3: Qualitative comparison of ELSA and LaSP on two challenging queries. ELSA produces programs that better reflect the true referential intent: correctly identifying the target category and using semantically appropriate spatial relations. Green boxes indicate predicted objects.

Table 4: Sr3D grounding accuracy (%, GT bounding boxes). **Bold**: best result.

Method	Overall
SAT (Yang et al., 2021)	57.9
MVT (Huang et al., 2022)	64.5
BUTD-DETR (Jain et al., 2022)	67.0
ViL3DRef (Chen et al., 2022)	72.8
3D-VisTA (Zhu et al., 2023)	76.4
Transcrib3D (Fang et al., 2024)	96.5
ELSA-1.7B (Ours)	97.2
ELSA-4B (Ours)	**97.8**

Table 5: Ablation study on Nr3D GT-label overall accuracy (%).

Configuration	Overall
Effect of RL training	
Qwen3-4B (zero-shot, no RL)	58.00
+ GRPO training (ELSA-4B)	**69.31**
Reward design	
Binary reward (0/1)	68.52
w/o alignment reward	68.95
Full reward (Ours)	**69.31**
Model scale	
ELSA-1.7B	66.68
ELSA-4B	**69.31**

absent. Both components contribute modestly but consistently to the full reward.

Model scale. ELSA-1.7B (66.68%) and ELSA-4B (69.31%) both surpass the GPT-4o baseline (65.7%), with a 2.6-point gap between scales. Surpassing a much larger LLM does not require a large SLM, demonstrating that execution-feedback RL transfers effectively across model sizes.

4.5 Analysis

Training Dynamics. Figure 4 shows reward and response length throughout GRPO training for both model scales. **Reward** (Figure 4a): ELSA-4B converges faster, reaching a stable reward of ≈0.8 within 500 steps, while ELSA-1.7B requires ≈1,500 steps to reach a comparable level. Both curves are smooth and monotonically increasing, with no signs of training collapse. **Response length** (Figure 4b): both models progressively reduce their output from ≈75 to ≈60 tokens over training. This shortening reflects the model learning to generate more targeted symbolic programs with fewer redundant relations and tighter category specifications, rather than verbose outputs that may confuse the execution engine.

Data Efficiency. Evaluating checkpoints on Nr3D, ELSA-4B surpasses the GPT-4o-based LaSP baseline (65.7%) after only 350 RL steps (≈2,800 training samples). The rapid convergence is consistent with the fast reward growth seen in Fig-

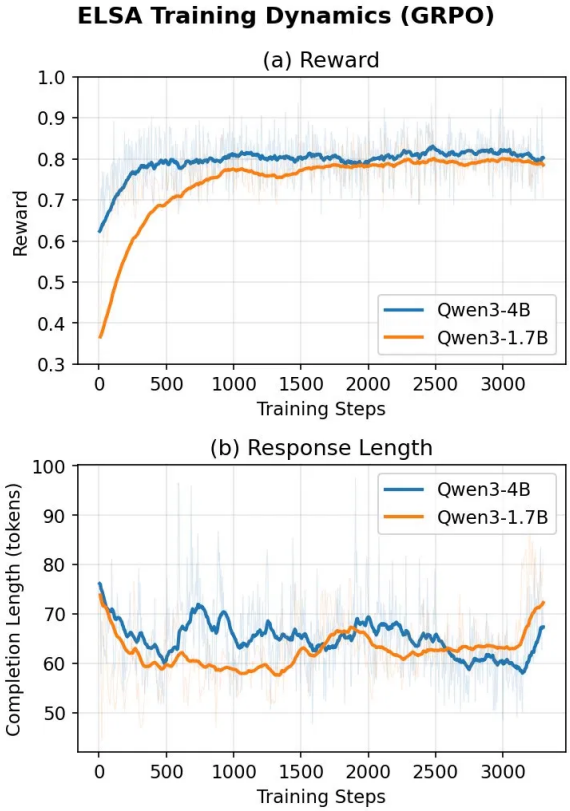

Figure 4: ELSA training dynamics for Qwen3-4B and Qwen3-1.7B. **(a)** Reward increases smoothly for both scales; 4B converges faster (≈500 steps) than 1.7B (≈1,500 steps). **(b)** Response length decreases over training, indicating that models learn to generate more concise symbolic programs.

ure 4a: the model requires fewer than 500 steps to stabilise, suggesting that spatial relational reasoning is quickly elicited from execution feedback rather than requiring extensive training. This makes ELSA practical to retrain and redeploy on new 3DVG environments without large-scale data collection.

View-Dependent Improvement. ELSA-4B narrows the VD–VI accuracy gap from 10.4 points (LaSP: 58.7% VD vs. 69.1% VI) to 8.49 points (ELSA-4B: 63.45% vs. 71.94%). Without any explicit frame-of-reference supervision, execution feedback implicitly teaches SLMs to produce more viewpoint-aware spatial programs, as incorrect egocentric interpretations consistently receive low execution rewards.

5 Conclusion

ELSA shows that execution outcomes from a symbolic grounding engine provide a sufficient training signal for SLMs to learn spatial language parsing without any intermediate annotation. Trained on Nr3D alone, our 1.7B and 4B models surpass GPT-4o-based LaSP on Nr3D, ScanRefer, and Sr3D at a fraction of the inference cost, with the 4B model reaching a competitive accuracy level after fewer than 3,000 training samples. The data efficiency and cross-dataset consistency suggest that execution feedback directly targets the representational bottleneck in spatial language parsing, rather than simply fitting dataset-specific patterns. Whether similar execution-feedback pipelines generalise to other grounded language tasks where deterministic verification is available remains an open question.

Limitations

ELSA's performance is bounded in part by LaSP's relation encoders, which compute direction relations from a fixed global coordinate origin rather than the observer's viewpoint; RL training partially mitigates the resulting errors on view-dependent queries but does not eliminate them. As a purely linguistic method, ELSA cannot resolve queries that require visual attribute verification, where grounding depends on perception beyond 3D bounding box coordinates. All experiments are conducted on ScanNet indoor scenes.

Ethical Considerations

This work trains models on publicly available 3D vision-language datasets (Nr3D, ScanRefer, Sr3D), which are derived from the ScanNet indoor scene dataset. These datasets contain no personally identifiable information. Our models process referring expressions and 3D object coordinates; no private or sensitive data is involved at any stage of training or evaluation.

The primary application domain of 3D visual grounding is embodied AI, including robotics and AR/VR assistants. While we do not foresee immediate harmful uses of spatial grounding technology, deployment in physical environments raises standard safety considerations for any robot or autonomous system acting on language instructions.

Compared to prior methods that require GPT-4o API calls at inference time, ELSA reduces both monetary cost and energy consumption by running a locally deployed SLM. The training compute is modest relative to large-scale pretraining.

References

Eslam Abdelrahman, Mohamed Ayman Mohamed, Mahmoud Ahmed, and Mohamed Elhoseiny. 2024. Cot3dref: Chain-of-thoughts data-efficient 3d visual grounding. In *International Conference on Learning Representations*, volume 2024, pages 11871–11896.

Panos Achlioptas, Ahmed Abdelreheem, Fei Xia, Mohamed Elhoseiny, and Leonidas Guibas. 2020. Referit3d: Neural listeners for fine-grained 3d object identification in real-world scenes. In *European conference on computer vision*, pages 422–440. Springer.

Dave Zhenyu Chen, Angel X Chang, and Matthias Nießner. 2020. Scanrefer: 3d object localization in rgb-d scans using natural language. In *European conference on computer vision*, pages 202–221. Springer.

Shizhe Chen, Pierre-Louis Guhur, Makarand Tapaswi, Cordelia Schmid, and Ivan Laptev. 2022. Language conditioned spatial relation reasoning for 3d object grounding. *Advances in neural information processing systems*, 35:20522–20535.

Nikita Drozdov, Andrey Lemeshko, Nikita Gavrilov, Anton Konushin, Danila Rukhovich, and Maksim Kolodiazhnyi. 2026. Z3d: Zero-shot 3d visual grounding from images. *arXiv preprint arXiv:2602.03361*.

Jiading Fang, Xiangshan Tan, Shengjie Lin, Igor Vasiljevic, Vitor Guizilini, Hongyuan Mei, Rares Ambrus, Gregory Shakhnarovich, and Matthew R Walter. 2024. Transcrib3d: 3d referring expression resolution through large language models. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 9737–9744. IEEE.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Haifeng Huang, Yilun Chen, Zehan Wang, Rongjie Huang, Runsen Xu, Tai Wang, Luping Liu, Xize Cheng, Yang Zhao, Jiangmiao Pang, and 1 others. 2024. Chat-scene: Bridging 3d scene and large language models with object identifiers. *Advances in Neural Information Processing Systems*, 37:113991–114017.

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

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

Zhao Jin, Rong-Cheng Tu, Jingyi Liao, Wenhao Sun, Xiao Luo, Shunyu Liu, and Dacheng Tao. 2026. Spazer: Spatial-semantic progressive reasoning agent for zero-shot 3d visual grounding. *Advances in Neural Information Processing Systems*, 38:165549–165576.

Rong Li, Shijie Li, Lingdong Kong, Xulei Yang, and Junwei Liang. 2025. Seeground: See and ground for zero-shot open-vocabulary 3d visual grounding. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 3707–3717.

Chen Liang, Mohammad Norouzi, Jonathan Berant, Quoc V Le, and Ni Lao. 2018. Memory augmented policy optimization for program synthesis and semantic parsing. *Advances in Neural Information Processing Systems*, 31.

Yuqi Liu, Bohao Peng, Zhisheng Zhong, Zihao Yue, Fanbin Lu, Bei Yu, and Jiaya Jia. 2025a. Seg-zero: Reasoning-chain guided segmentation via cognitive reinforcement. *arXiv preprint arXiv:2503.06520*.

Yuqi Liu, Tianyuan Qu, Zhisheng Zhong, Bohao Peng, Shu Liu, Bei Yu, and Jiaya Jia. 2025b. Vision-reasoner: Unified reasoning-integrated visual perception via reinforcement learning. *arXiv preprint arXiv:2505.12081*.

Boyu Mi, Hanqing Wang, Tai Wang, Yilun Chen, and Jiangmiao Pang. 2025. Language-to-space programming for training-free 3d visual grounding. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 3844–3864.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, and 1 others. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Yanmin Wu, Xinhua Cheng, Renrui Zhang, Zesen Cheng, and Jian Zhang. 2023. Eda: Explicit text-decoupling and dense alignment for 3d visual grounding. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 19231–19242.

Runsen Xu, Zhiwei Huang, Tai Wang, Yilun Chen, Jiangmiao Pang, and Dahua Lin. 2024. Vlm-grounder: A vlm agent for zero-shot 3d visual grounding. *arXiv preprint arXiv:2410.13860*.

Jianing Yang, Xuweiyi Chen, Shengyi Qian, Nikhil Madaan, Madhavan Iyengar, David F Fouhey, and Joyce Chai. 2024. Llm-grounder: Open-vocabulary

3d visual grounding with large language model as an agent. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7694–7701. IEEE.

Zhengyuan Yang, Songyang Zhang, Liwei Wang, and Jiebo Luo. 2021. Sat: 2d semantics assisted training for 3d visual grounding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1856–1866.

Qihao Yuan, Kailai Li, and Jiaming Zhang. 2024a. Solving zero-shot 3d visual grounding as constraint satisfaction problems. *arXiv preprint arXiv:2411.14594*.

Zhihao Yuan, Jinke Ren, Chun-Mei Feng, Hengshuang Zhao, Shuguang Cui, and Zhen Li. 2024b. Visual programming for zero-shot open-vocabulary 3d visual grounding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20623–20633.

Nader Zantout, Haochen Zhang, Pujith Kachana, Jinkai Qiu, Guofei Chen, Ji Zhang, and Wenshan Wang. 2025. Sort3d: Spatial object-centric reasoning toolbox for zero-shot 3d grounding using large language models. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2201–2208. IEEE.

Tatiana Zemskova and Dmitry Yudin. 2025. 3dgraphllm: Combining semantic graphs and large language models for 3d scene understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8885–8895.

Shengli Zhou, Minghang Zheng, Feng Zheng, and Yang Liu. 2026. Scalable object relation encoding for better 3d spatial reasoning in large language models. *arXiv preprint arXiv:2603.24721*.

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

A Implementation Details

Models. We train two model variants: **Qwen3-1.7B** and **Qwen3-4B**, both fine-tuned with LoRA (rank $= 16$, $\alpha = 32$, dropout $= 0.05$) applied to all attention projections (`q/k/v/o_proj`) and feed-forward projections (`gate/up/down_proj`). Training uses the `trl` library's `GRPOTrainer`.

Training configuration. We train on the full Nr3D training split (26,616 samples) with $G = 8$ rollouts per query. Key hyperparameters:

No ScanRefer or Sr3D data is used during training; both are evaluated in a zero-shot transfer setting.

Hyperparameter	ELSA-4B	ELSA-1.7B
Learning rate	8×10^{-5}	8×10^{-5}
Per-device batch size	8	16
Gradient accumulation	1	1
Group size G	8	8
Temperature	0.7	0.7
Max completion length	512	512
Max prompt length	768	768
GPUs	8×H20	8×H20
Precision	bf16	bf16

Table 6: Training hyperparameters for both model scales.

Inference. At inference time, only the fine-tuned SLM runs; the LaSP engine and all geometric encoders remain frozen. When the SLM predicts a category absent from precomputed scene features, we resolve it on-the-fly via CLIP embeddings to ensure the engine can execute the program. Inference on 8×H20 GPUs takes 0.06 s per query.

LaSP engine. We use LaSP's official codebase with its pre-generated Python relation encoders. ELSA does not modify any encoder weights or logic; the only change is replacing GPT-4o with the fine-tuned SLM for query parsing.

B Reward Design Details

The reward function has three levels applied in order. Only when a sample passes Level 1 does it proceed to Level 2 and Level 3.

Level 1: Format penalties. These detect malformed outputs before any execution:

Format violation	Penalty
JSON parse failure	-1.0
No relations present	-0.5
All relation names invalid	-0.3
Relation arity error $> 50\%$	-0.3

Table 7: Level 1 format penalties applied before execution.

Level 2: Alignment bonus. An additional bonus rewards semantic consistency between the predicted program's category and the relation anchors:

$$r = r_{\text{exec}} + \lambda \cdot r_{\text{align}}, \quad \lambda = 0.05 \tag{8}$$

Level 3: Execution reward. If the format is valid, the program is executed by the LaSP engine:

$$r_{\text{exec}} = \begin{cases} 1/\text{rank}(o^*) & \text{if rank}(o^*) \leq 5 \\ 0.2 & \text{otherwise} \end{cases} \tag{9}$$

where $\text{rank}(o^*)$ is the position of the ground-truth object in the engine's sorted output.

When execution fails entirely (e.g., missing scene data), the reward reduces to $0.1 \cdot r_{\text{align}}$ to still provide a gradient signal for structural correctness.

C Prompt Template

The SLM receives the following system prompt at both training and inference time:

```
You are a spatial relation parser. Convert a
referring expression into a structured JSON program.

## Output Format
{"category": "X", "relations": [{"relation_name":
    "R","objects": [...], "negative": false}]}

## Valid relation_names (ONLY use these):
- Unary (objects=[]):
    corner, on the floor, against wall, smaller,
    larger, taller, lower, within
- Binary (objects=[one object]):
   above, below, beside, close, far, left, right,
    front, behind, across
- Ternary (objects=[two objects]):
    between, center, middle

## Mapping rules:
- "on/on top of" -> above
- "under/beneath/below" -> below
- "next to/near/by/beside" -> close
- "not X" -> set "negative": true
- category = simple noun only (chair, table, bed).
  No adjectives/colors.

## Examples:

Input: "The chair closest to the door on the left side
of the room."
Output: {"category": "chair", "relations":
  [{"relation_name": "close",
   "objects": [{"category": "door", "relations": []}],
    "negative": false},
   {"relation_name": "left",
   "objects": [{"category": "door", "relations": []}],
    "negative": false}]}

Input: "The pillow between the two nightstands, not on
the bed."
Output: {"category": "pillow", "relations":
  [{"relation_name": "between",
   "objects": [{"category": "nightstand", "relations":[]},
           {"category": "nightstand", "relations":[]}],
    "negative": false},
   {"relation_name": "above",
   "objects": [{"category": "bed", "relations": []}],
    "negative": true}]}
```

The user message is simply: `Parse: "{utterance}"`

Note that the prompt does *not* include scene object lists or coordinates. The SLM produces a symbolic program purely from the natural language query; spatial grounding is performed entirely by the downstream LaSP engine using precomputed 3D features.

D Additional Qualitative Examples

We provide interactive 3D visualizations in the supplementary code archive (`ELSA/visualization/`). Each HTML file shows:

- The full ScanNet scene mesh with original vertex colors
- **Green** bounding box: ground-truth target object
- **Red** bounding box: LaSP baseline's incorrect prediction

These demonstrate cases where ELSA correctly grounds the target (rank = 1) while LaSP fails significantly (rank > 20). Open the HTML files in a browser for interactive 3D inspection.