EGOALIGN: SITUATED 3D UNDERSTANDING WITHOUT SITUATION DESCRIPTIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Situated 3D Question Answering requires models to interpret a 3D scene relative to an agent's pose. Existing methods commonly convey that pose through a natural-language situation description, which may be unavailable or costly to obtain in real-world deployments. While some methods additionally incorporate numerical pose, we find that randomly shuffling the agent pose causes little performance degradation, indicating limited functional dependence on the numerical pose. To address this gap, we introduce EgoAlign, a simple 3D language model that represents a scene in an agent-centric coordinate frame, together with a dedicated egocentric alignment stage comprising four complementary tasks: object category recognition, egocentric direction prediction, metric distance estimation, and spatial relation prediction. This stage teaches the model to consistently interpret scene geometry relative to the agent's pose before task-specific fine-tuning. EgoAlign achieves the strongest performance on SQA3D, Real-3DQA, and MSQA among compared methods while remaining highly efficient. More importantly, removing textual situation descriptions yields comparable or better downstream performance, whereas shuffling EgoAlign's numerical pose causes substantial degradation. These results show that EgoAlign establishes agent-centric spatial context directly from numerical pose without requiring textual situation descriptions.

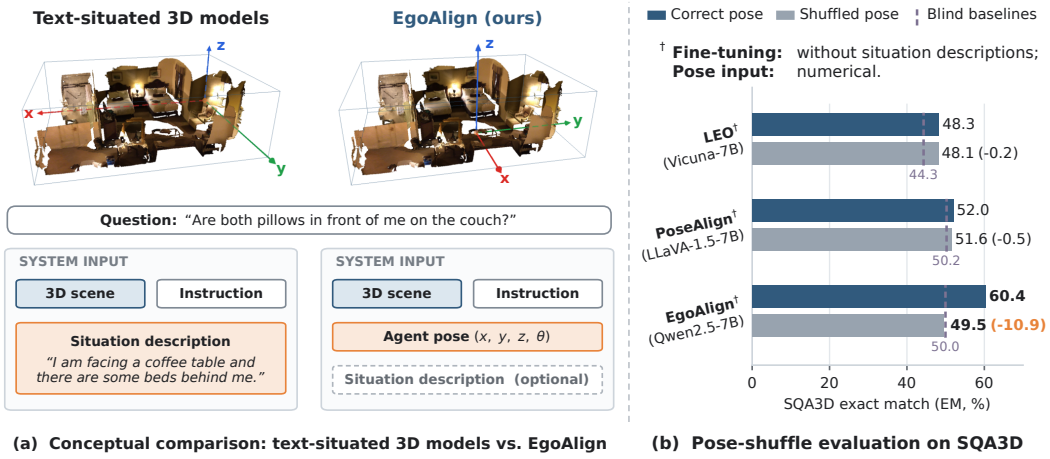


Figure 1: **Conceptual overview and pose-shuffle evaluation.** (a) Text-situated 3D models specify the agent's situation using a natural-language description, whereas EgoAlign constructs its spatial reference frame directly from numerical pose. (b) We evaluate pose-conditioned models after fine-tuning on SQA3D without situation descriptions. Blind baseline means model fine-tuned without 3D input. Shuffling poses has little effect on LEO and PoseAlign, but substantially degrades EgoAlign to near the blind-baseline level, showing that EgoAlign's predictions are strongly dependent on the supplied numerical pose.

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.

1 INTRODUCTION

Recent 3D language models have made substantial progress in grounding large language models (LLMs) in 3D environments, enabling tasks such as question answering, grounding, planning, and embodied reasoning (Hong et al., 2023; Zhu et al., 2023; Chen et al., 2024a; Huang et al., 2023; 2024; Deng et al., 2025; Zhi et al., 2025). For an embodied agent, understanding a scene also requires knowing where the agent is and which direction it faces: whether an object is to the left, in front, or behind depends on the agent's pose (Zhang et al., 2024). Situated 3D benchmarks such as SQA3D and MSQA capture this requirement by conditioning questions on an agent-specific situation (Ma et al., 2022; Linghu et al., 2024). Most existing methods represent this situation through natural-language or multimodal descriptions, a formulation that naturally leverages the strong language understanding capabilities of pretrained LLMs (Man et al., 2024; Zhang et al., 2024; Yuan et al., 2025).

However, for a physical agent, numerical pose is a more direct representation of its state. Localization systems naturally provide position and orientation (Cadena et al., 2016; Campos et al., 2021), whereas a situation description is an indirect linguistic rendering that must be regenerated as the agent moves (Ma et al., 2022; Liu et al., 2026). Ideally, a 3D language model should therefore establish its spatial reference frame directly from numerical pose instead of relying on textual situation descriptions. A few recent models have started to incorporate numerical pose, for example by encoding pose as an additional input or transforming scene geometry into an ego-aligned coordinate frame (Huang et al., 2023; Linghu et al., 2024; Zhang et al., 2024; Liu et al., 2026).

Yet **incorporating numerical pose is not the same as using it**. QA accuracy alone does not reveal whether a model actually relies on its numerical pose. We test pose dependence directly by shuffling numerical poses while keeping the scene and question unchanged. As shown in Figure 1, even after fine-tuning without situation descriptions, LEO and PoseAlign lose only 0.2 and 0.5 exact-match points, respectively. Their predictions are therefore largely insensitive to whether the supplied pose is correct; more details are provided in Appendix A. We refer to this failure mode as **pose insensitivity**: a model may receive numerical pose, or even transform its input geometry with it, while learning a prediction function that depends little on that pose.

To address this problem, we introduce **EgoAlign**, a point-cloud language model that makes numerical pose operational at both the representation and supervision levels. At the representation level, it transforms the scene into an agent-centric coordinate frame and explicitly preserves the metric coordinates associated with each point token, so that the LLM receives both semantic and geometric information relative to the current pose. At the supervision level, we introduce an egocentric alignment stage with four complementary tasks: object category recognition, egocentric direction prediction, metric distance estimation, and spatial relation prediction. For each scene configuration, we keep the scene, agent position, and referred objects fixed while varying only the numerical heading across four orientations, creating counterfactual groups that explicitly teach the model how spatial interpretation should change with the agent frame.

EgoAlign achieves the strongest performance on SQA3D, Real-3DQA, and MSQA among compared methods while remaining computationally efficient. More importantly, removing textual situation descriptions yields comparable or better downstream performance, showing that the model can establish its spatial context directly from numerical pose. Under the same pose-shuffle intervention on SQA3D, EgoAlign drops around 10 points to its blind baseline level, while existing pose-conditioned models change little. Its performance also degrades smoothly under increasing yaw perturbations. Together, these results show that EgoAlign is not only accurate but functionally dependent on the numerical pose it receives. Our contributions are:

- We identify **pose insensitivity** as a hidden failure mode of situated 3D language models.
- We propose **EgoAlign**, an agent-centric point-cloud language model that explicitly preserves metric coordinates at the point-token level.
- We introduce an **egocentric alignment** stage with multi-heading counterfactual supervision that teaches the model to interpret geometry relative to numerical pose.
- Across SQA3D, Real-3DQA, and MSQA, EgoAlign demonstrate superior performance without requiring textual situation descriptions and exhibits strong pose dependence.

2 Related Work

Language models for 3D scene understanding. Recent 3D language models connect pretrained LLMs with point-cloud or object-level scene representations, enabling tasks such as question answering, captioning, grounding, planning, and embodied interaction (Hong et al., 2023; Zhu et al., 2023; Wang et al., 2023; Xu et al., 2024). These methods mainly focus on grounding language in scene content, i.e., recognizing what objects and structures are present and reasoning about their relations (Chen et al., 2024b; Huang et al., 2024; Mei et al., 2025). EgoAlign follows this general point-cloud language modeling paradigm, but focuses on a complementary aspect: representing the same 3D scene relative to a changing agent pose. To support such reasoning, we explicitly preserve metric coordinates in the agent-centric frame instead of relying only on compressed semantic features.

Situated 3D understanding. Situated 3D understanding extends scene reasoning to a particular agent position and orientation. SQA3D introduces this setting using natural-language situation descriptions, while subsequent works such as SIG3D, SPARTUN3D, and MSQA improve situation grounding, spatial alignment, and multimodal situation modeling (Ma et al., 2022; Man et al., 2024; Zhang et al., 2024; Linghu et al., 2024). Most methods still specify the agent state primarily through descriptive language, although a smaller set of models, including LEO, MSR3D, and PoseAlign, also incorporate numerical pose (Huang et al., 2023; Linghu et al., 2024; Liu et al., 2026). EgoAlign differs by treating numerical pose as the primary reference frame and explicitly training the model to make pose-dependent predictions through egocentric alignment supervision.

3 Method

3.1 Preliminary

Let a 3D scene be a colored point cloud $\mathcal{P} = \{(\mathbf{p}_i, \mathbf{c}_i)\}_{i=1}^{N}$ in the scan's default coordinate system, where $\mathbf{p}_i = (x_i, y_i, z_i)$ and $\mathbf{c}_i = (r_i, g_i, b_i)$ are the XYZ coordinates and RGB color of point i, respectively. For an agent pose with position $\mathbf{t} = (x, y, z)$ and heading θ, we first transform $\mathcal{P}$ into $\mathcal{P}_{\text{ego}}$, where the agent is at the origin with $+x$ forward, $+y$ left, and $+z$ up. Given an instruction q and an optional situation description s (or $s = \varnothing$ when unavailable), the model predicts

$$\hat{a} = f_\phi(\mathcal{P}_{\text{ego}}, q, s), \quad \mathcal{P}_{\text{ego}} = \{(\widetilde{\mathbf{p}}_i, \mathbf{c}_i)\}_{i=1}^{N}, \quad \widetilde{\mathbf{p}}_i = R_z(-\theta)(\mathbf{p}_i - \mathbf{t}). \tag{1}$$

3.2 Point-cloud language model

EgoAlign connects a pretrained point encoder to an LLM. The encoder compresses the transformed cloud into $M \ll N$ semantic tokens $\{\mathbf{z}_j\}_{j=1}^{M}$. Although the input cloud is expressed in the agent-centric frame, the compressed semantic tokens do not explicitly preserve the metric location of their supporting points. This information can be attenuated by point-cloud encoders designed to learn robust, translation and rotation invariant geometric representations (Wu et al., 2025; Zhang et al., 2026), yet it is essential for reasoning about the scene relative to the agent.

To address this problem, we infuse metric coordinates explicitly. For token j, we track the encoder's pooling assignments to obtain $\mathcal{I}_j$, the original input points contributing to that token. We mean-pool the coordinates and colors of the points in $\mathcal{I}_j$, then project them alongside the semantic feature into the LLM's embedding space:

$$\mathbf{e}_j = \frac{1}{\sqrt{3}} \left[f_{\text{sem}}(\mathbf{z}_j) + f_{\text{coord}}\left(\operatorname*{avg}_{i \in \mathcal{I}_j} \widetilde{\mathbf{p}}_i \right) + f_{\text{rgb}}\left(\operatorname*{avg}_{i \in \mathcal{I}_j} \mathbf{c}_i \right) \right]. \tag{2}$$

We scale the sum by $1/\sqrt{3}$ to approximately preserve the feature variance when combining three independently projected branches. The fused point cloud embeddings are then processed by the LLM's attention layers together with the regular text embeddings.

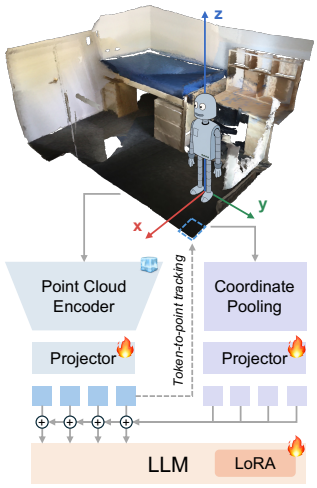


Figure 2: EgoAlign architecture with explicit coordinate fusion. The color branch is omitted for clarity.

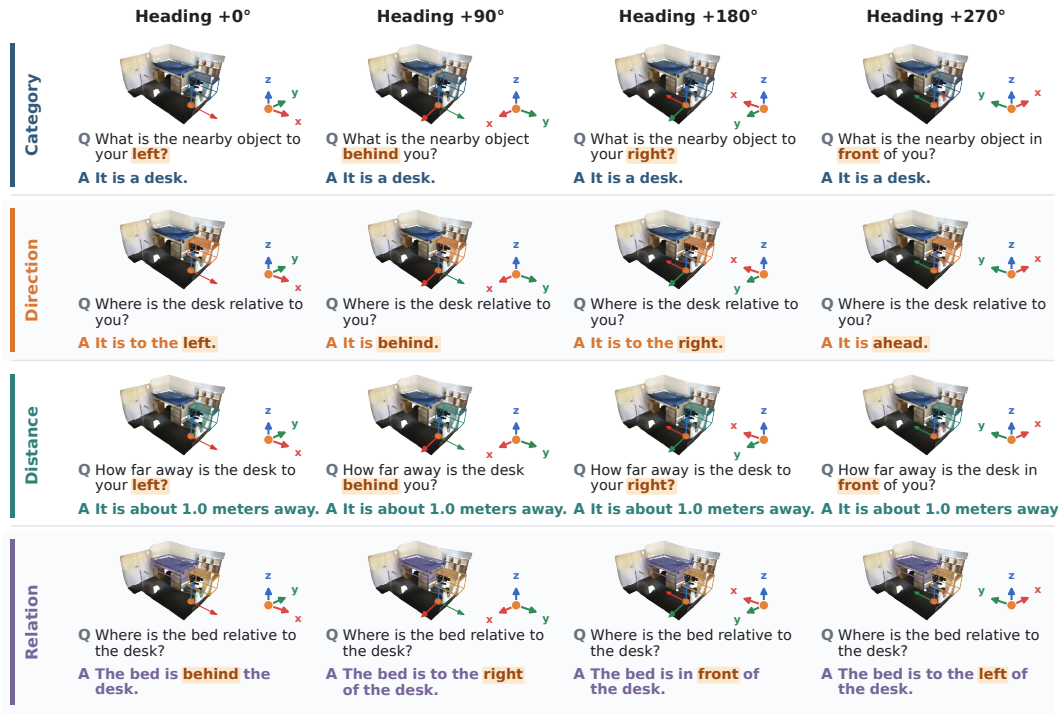


Figure 3: Egocentric alignment supervision at a fixed scene and position. Rotating the numerical heading changes agent-relative questions and answers; the scene geometry is unchanged.

3.3 Egocentric Alignment

A suitable representation alone does not ensure that the model learns to use its numerical reference frame. Downstream situated QA also provides insufficient supervision for learning a pose-dependent spatial representation. Besides the linguistic shortcuts present in existing benchmarks (Huang et al., 2025; Ma et al., 2026), the training data contain very limited variation in heading at a fixed agent position. In SQA3D, the 26,623 training questions correspond to 5,308 unique agent positions, of which 5,212 appear with only a single heading. As a result, the model rarely observes how the same scene configuration should be interpreted differently when only the agent orientation changes. We therefore introduce a dedicated egocentric alignment stage that directly supervises the mapping between scene geometry and the agent-centric reference frame. We construct the alignment data from scenes and agent positions in the SQA3D training split, using four complementary tasks:

- **Category** asks the model to identify an object from its color and egocentric location, grounding semantic point tokens to a specified spatial region.
- **Direction** predicts one of eight horizontal directions from the agent to an object, explicitly supervising how object location changes with the agent heading.
- **Distance** estimates the nearest horizontal distance from the agent to an object's box, rounded to 0.1 meters, connecting egocentric coordinates to quantitative spatial language.
- **Relation** describes one object's location relative to another using eight agent-centric horizontal relations together with the viewpoint-invariant *above* and *below* relations.

Crucially, each query is expanded into a four-heading counterfactual group. We keep the scene, agent position, and referred objects fixed while instantiating headings at $0°$, $90°$, $180°$, and $270°$. Thus, the underlying scene content remains unchanged, while the egocentric coordinates and corresponding spatial labels must vary consistently with the numerical heading. This construction provides the heading variation largely missing from downstream QA and directly teaches the model how spatial interpretation should transform with the agent frame. We sample 75K examples for each task,

yielding a balanced 300K-example alignment corpus, and jointly train all four tasks with the same autoregressive objective used for downstream QA. Figure 3 illustrates one four-heading group.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Model. EgoAlign couples a Concerto point encoder (Zhang et al., 2026) with a Qwen2.5-Instruct language model (Team, 2024). We instantiate EgoAlign with the 1.5B, 3B, and 7B variants of Qwen2.5-Instruct. During both egocentric alignment and downstream fine-tuning, the point encoder and base LLM weights remain frozen; we train only the projection layers and LoRA adapters. Optimization and implementation details are provided in Appendix B.

Datasets and metrics. We evaluate on SQA3D, Real-3DQA, and MSQA. SQA3D contains 33.4K situated questions across 650 ScanNet scenes. Real-3DQA is built on SQA3D to reduce linguistic shortcuts: it retains 1,485 test questions after filtering those answerable without 3D evidence, and augments 750 questions with four agent headings. MSQA contains 251K situated QA pairs across 1,734 scenes from ScanNet, 3RScan, and ARKitScenes. We report exact match (EM) and refined exact match (EM-R) on SQA3D and Real-3DQA. For the four-heading subset of Real-3DQA, we additionally report viewpoint robustness score (VRS), which averages QA accuracy across the four heading variants. On MSQA, we report scores using its official LLM-based evaluator.

Baselines. We compare EgoAlign with a broad set of 3D language models, including 3D-LLM (Hong et al., 2023), LEO (Huang et al., 2023), Chat-Scene (Huang et al., 2024), Spartun3D-LLM (Zhang et al., 2024), 3D-LLaVA (Deng et al., 2025), PoseAlign (Liu et al., 2026), MSR3D (Linghu et al., 2024), SceneCOT (Linghu et al., 2026), Scene-LLM (Fu et al., 2024), LLaVA-3D (Zhu et al., 2024), Video-3D LLM (Zheng et al., 2025), GPT4Scene (Qi et al., 2026), and Qwen-3D (Lin et al., 2026). We distinguish methods that consume 3D scene representations from those solely based on images or videos in the result tables.

4.2 DOWNSTREAM 3D QUESTION ANSWERING

SQA3D and Real-3DQA. Table 1 reports the main results on SQA3D and Real-3DQA using EgoAlign with Qwen2.5-7B. EgoAlign achieves the best overall performance on both benchmarks, reaching 61.2 EM and 64.3 EM-R on SQA3D and outperforming prior methods across all reported Real-3DQA metrics. Notably, strong performance does not require textual situation descriptions: removing them only slightly reduces SQA3D EM from 61.2 to 60.4, while slightly improving performance on Real-3DQA. These results show that EgoAlign can establish an effective egocentric reference frame directly from numerical pose, with situation text serving as optional auxiliary context rather than a necessary representation of the agent state.

Efficiency. EgoAlign is also substantially more efficient than existing high-performing models. As shown in Figure 4, the 7B model achieves 61.2 SQA3D EM with a batch-1 latency of only 0.40 s/question on an A40 GPU. It is simultaneously faster and more accurate than the compared methods, including both point-based and vision/video-based models. This efficiency comes from operating directly on a compact set of point tokens rather than processing multiple rendered views or video frames.

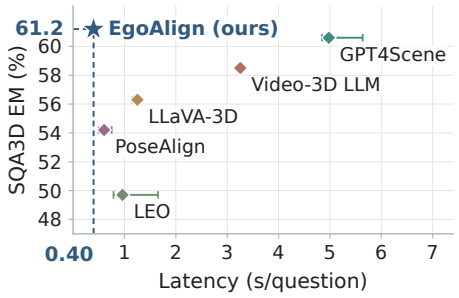


Figure 4: SQA3D exact match versus batch-1 latency. Markers and whiskers show median and interquartile. All methods are measured on A40 GPUs.

MSQA. Table 2 shows similar gains on the larger and more diverse MSQA benchmark. EgoAlign achieves an overall score of 58.5 with situation descriptions and 60.0 without them. The no-situation model performs particularly well on existence and spatial questions, reaching 93.0 and 54.0, respectively.

Table 1: Results on SQA3D and Real-3DQA. EgoAlign uses Qwen2.5-7B. Light-blue cells indicate methods that incorporate numerical poses. +sit./−sit. denote with/without situation descriptions.

Method	Encoder		SQA3D		Real-3DQA		
	Point	Vision	EM	EM-R	EM	EM-R	VRS
3D-LLM [NeurIPS 2023]	✓	✓	47.8	49.6	7.5	10.4	9.6
LEO [ICML 2024]	✓	✓	49.7	52.3	14.3	19.1	14.3
Chat-Scene [NeurIPS 2024]	✓	✓	54.4	57.2	17.0	22.1	12.9
Spartun3D-LLM [ICLR 2025]	✓	✓	54.8	–	–	–	–
3D-LLaVA [CVPR 2025]	✓	✗	54.5	56.6	19.7	24.9	14.8
PoseAlign [CVPR 2026]	✓	✗	54.2	56.9	18.3	23.2	14.9
Scene-LLM [WACV 2025]	✗	✓	54.2	–	–	–	–
LLaVA-3D [ICCV 2025]	✗	✓	56.3	59.0	27.7	32.6	18.3
Video-3D LLM [CVPR 2025]	✗	✓	58.5	61.2	32.3	37.3	15.2
GPT4Scene [ICLR 2026]	✗	✓	60.6	63.3	33.1	36.9	18.2
Qwen-3D [ECCV 2026]	✗	✓	59.6	–	–	–	–
EgoAlign (+sit.)	✓	✗	**61.2**	**64.3**	33.3	38.5	18.6
EgoAlign (−sit.)	✓	✗	60.4	63.2	**33.7**	**38.6**	**19.1**

Table 2: Results on MSQA. EgoAlign uses Qwen2.5-7B. Light-blue cells indicate methods that explicitly incorporate numerical poses. +sit./−sit. denote with/without situation descriptions.

Method	Encoder		MSQA						
	Point	Vision	Count.	Exist.	Attr.	Spatial	Nav.	Other	Overall
GPT-4o	–	–	31.2	71.4	**75.2**	31.5	36.7	**88.0**	49.7
LEO [ICML 2024]	✓	✓	32.5	88.5	58.7	44.2	39.6	81.4	54.8
Chat-Scene [NeurIPS 2024]	✓	✓	37.4	92.0	49.0	47.0	58.3	83.7	56.6
MSR3D [NeurIPS 2024]	✓	✓	33.5	87.5	53.7	48.9	**61.9**	75.0	56.5
SceneCOT [ICLR 2026]	✓	✓	**47.9**	82.1	49.6	47.2	51.6	80.3	55.6
EgoAlign (+sit.)	✓	✗	36.7	91.7	51.7	51.9	59.5	84.1	58.5
EgoAlign (−sit.)	✓	✗	37.0	**93.0**	54.7	**54.0**	61.2	83.1	**60.0**

The improvement after removing situation descriptions is consistent with the Real-3DQA results and suggests that direct numerical-pose grounding transfers beyond SQA3D to different datasets and scene sources.

4.3 Ablations

Table 3 studies the contributions of the agent-centric representation and egocentric alignment using Qwen2.5-1.5B. Removing alignment substantially degrades downstream accuracy and nearly eliminates sensitivity to pose shuffling, showing that downstream QA supervision alone is insufficient to establish a pose-dependent representation. Using the scan's default coordinates similarly harms both alignment and downstream performance. Removing the explicit coordinate projection causes a smaller accuracy drop, but also reduces Δ_{pose}, indicating that the agent-centric transform and explicit metric coordinates provide complementary signals for both reasoning accuracy and pose dependence. Alignment dynamics for these variants are provided in Appendix C.1.

The four alignment tasks provide complementary supervision. To isolate task composition, each task ablation resamples from the remaining tasks to keep the alignment set fixed at 300K examples. Category supervision is the most critical, while removing direction, distance, or relation causes smaller but consistent losses in downstream accuracy and pose sensitivity. The relatively mild effect of removing relation supervision is expected, as relation prediction is a higher-level task that overlaps with the directional and metric skills learned elsewhere. We also replace the four-heading counterfactual construction with a 300K-example single-heading variant. Although single-heading

Table 3: Ablations with Qwen2.5-1.5B. Models are fine-tuned and evaluated without situation descriptions. Distance is measured by MAE. Δ_{pose} is the decrease in EM under pose shuffling.

	Alignment				SQA3D		Real-3DQA	
Setting	Cat. ↑	Dir. ↑	Dist. ↓	Rel. ↑	EM ↑	Δ_{pose}	EM ↑	Δ_{pose}
Full	**61.8**	**59.0**	**0.47**	**42.2**	**56.8**	9.72	**30.2**	10.30
No alignment	–	–	–	–	49.0	0.14	14.8	0.07
Default coordinates	28.5	12.5	1.15	11.6	48.6	–	16.4	–
No coordinate proj.	49.5	38.4	0.86	34.2	55.8	6.17	25.9	6.46
No category task	0.3	54.0	0.73	27.2	51.3	2.13	18.1	1.41
No direction task	47.7	23.6	1.01	30.6	53.0	3.24	23.5	5.19
No distance task	55.9	45.3	–	36.5	55.8	6.25	23.6	7.01
No relation task	61.5	56.2	0.48	6.8	56.0	7.72	27.8	8.16
Single heading	–	–	–	–	55.6	7.13	27.0	7.28

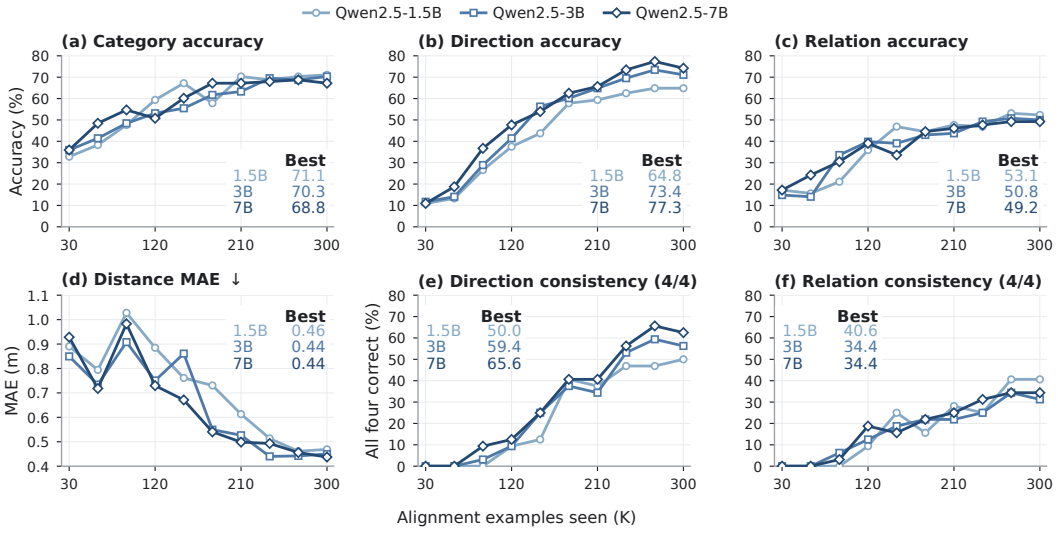


Figure 5: Alignment dynamics across Qwen2.5 model scales.

alignment remains beneficial, it consistently underperforms the full model in both downstream accuracy and Δ_{pose}. Thus, single-heading supervision is already useful, while explicit heading variation provides additional supervision for learning a stronger pose-dependent representation.

5 Analysis

5.1 Effect of model scale

Figure 5 examines how egocentric alignment evolves across Qwen2.5 model scales. All three models improve steadily as more alignment examples are observed, although scaling affects individual skills differently. Larger language models benefit direction prediction most clearly, with the best accuracy increasing from 64.8 for 1.5B to 77.3 for 7B, while category and relation accuracy do not consistently improve with model size. Importantly, four-heading consistency also rises markedly during alignment, requiring the model to answer all four counterfactual heading variants correctly. This shows that EgoAlign learns to update its spatial interpretation consistently with the numerical heading. Overall, increasing language-model capacity appears particularly beneficial for heading-dependent directional reasoning, while some grounding skills depend more strongly on the shared 3D representation.

Figure 6 shows the corresponding downstream results. SQA3D generally improves with model scale, while Real-3DQA also benefits most from the 7B model. MSQA improves from 1.5B to 3B and then largely saturates. Across all three scales, removing situation descriptions improves

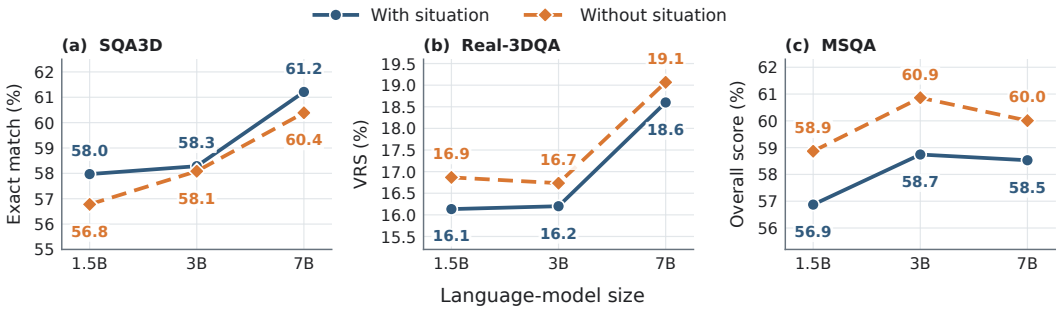


Figure 6: Downstream performance across Qwen2.5 model scales with and without situation descriptions. Complete results are in Appendix C.2.

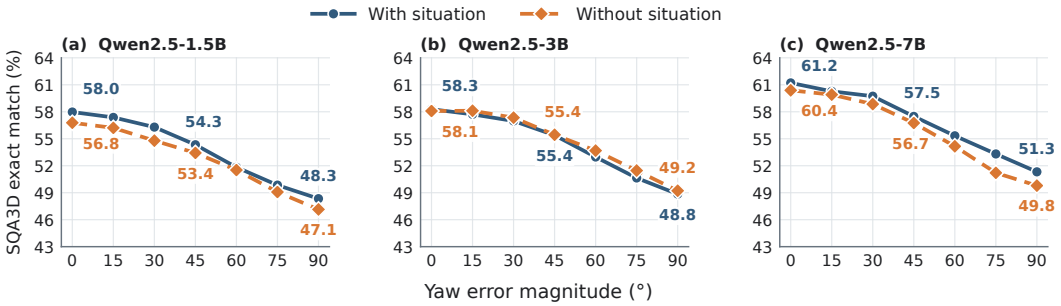


Figure 7: SQA3D exact match under numerical yaw error across Qwen2.5 model scales. Perturbation signs are balanced, and 0° denotes the clean pose. Exact values are in Appendix C.3.

performance on Real-3DQA and MSQA, whereas SQA3D retains a small benefit from situation text. Thus, EgoAlign does not rely on increasing model scale to compensate for missing situation descriptions; the no-situation setting remains competitive across the full range of model sizes.

5.2 Sensitivity to pose error

A model that genuinely depends on numerical pose should degrade predictably when that pose becomes inaccurate. We therefore perturb the input yaw by increasing angular errors while keeping the scene and question unchanged. Figure 7 shows a clear degradation across all model scales: for the 7B model, SQA3D EM decreases from 61.2 at the correct pose to 51.3 at a 90° error with situation descriptions, and from 60.4 to 49.8 without them. Similar trends appear for the 1.5B and 3B variants.

The degradation is gradual rather than abrupt, linking model performance directly to the fidelity of the supplied reference frame. Situation descriptions provide only limited robustness to pose corruption: the trends with and without situation text remain closely matched. This indicates that EgoAlign continues to rely primarily on its numerical pose even when linguistic situation context is available.

5.3 Can alignment supervision transfer to existing architectures?

One possibility is that our alignment data alone is sufficient to make any pose-conditioned 3D language model pose-sensitive. We test this by applying the same 300K egocentric alignment examples to 7B LEO and PoseAlign variants. As shown in Table 4, both models learn the alignment tasks substantially less effectively than EgoAlign. More importantly, their downstream predictions remain insensitive to pose: shuffling the numerical pose changes SQA3D EM by $+0.06$ for LEO and $+0.28$ for PoseAlign, compared with a -10.88 point drop for EgoAlign.

These results indicate that alignment supervision alone is insufficient. Effective pose grounding requires a representation that exposes agent-centric metric geometry to the language model together with supervision that teaches how this geometry should be interpreted. EgoAlign is designed around both components, whereas simply applying the same alignment data to existing architectures does not produce comparable pose dependence.

Table 4: Transfer of egocentric alignment supervision to existing 7B models. All methods use the same 300K alignment examples; SQA3D uses no-situation fine-tuning and evaluation. Distance is MAE in meters; Δ_{pose} is the decrease in EM under pose shuffling.

	Alignment validation				SQA3D EM			SQA3D EM-R		
Method	Cat. ↑	Dir. ↑	Dist. ↓	Rel. ↑	Correct	Shuffled	Δ_{pose}	Correct	Shuffled	Δ_{pose}
LEO	30.9	12.7	0.95	12.5	46.63	46.69	−0.06	48.59	48.57	0.03
PoseAlign	36.3	13.0	1.09	13.2	50.13	50.41	−0.28	52.46	52.71	−0.26
EgoAlign	**63.2**	**61.3**	**0.42**	**46.6**	**60.39**	**49.50**	10.88	**63.23**	**52.12**	11.11

5.4 Qualitative Analysis

Figure 8 shows two multi-heading examples from Real-3DQA. PoseAlign tends to keep nearly the same answer as the heading changes, while EgoAlign adapts its prediction to the agent frame. In the second example, the dataset answer at 180° is incorrect: the object to the agent's left is an armchair, which is also predicted by EgoAlign. Overall, the examples illustrate that EgoAlign responds consistently to changes in numerical heading rather than producing pose-invariant answers. Additional multi-heading case studies are provided in Appendix C.4.

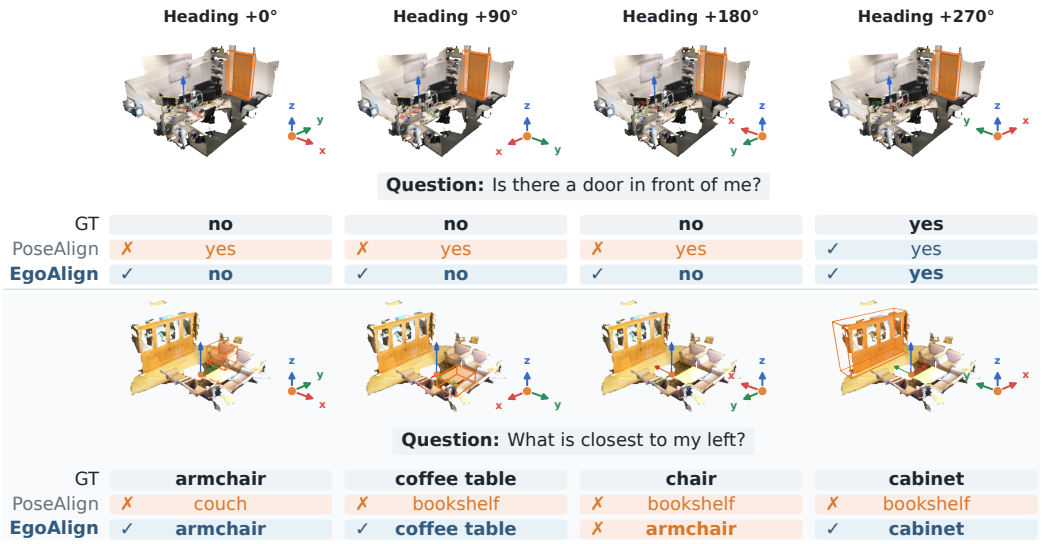


Figure 8: Qualitative Real-3DQA comparisons across four annotated headings.

6 Conclusion

We presented EgoAlign, a 3D language model for situated reasoning directly from numerical agent pose. Our experiments reveal pose insensitivity as an important failure mode of existing pose-conditioned models: even when textual situation description is unavailable, their predictions can remain largely unchanged when the numerical pose is corrupted. EgoAlign addresses this issue through an agent-centric point representation that preserves metric coordinates and a dedicated egocentric alignment stage that explicitly teaches how spatial interpretation changes with the agent frame. Across SQA3D, Real-3DQA, and MSQA, EgoAlign achieves excellent performance while remaining efficient, and performs comparably or better without textual situation descriptions. Its predictions also degrade consistently under pose shuffling and increasing yaw error, demonstrating substantially stronger dependence on the supplied reference frame. These results suggest that numerical pose can serve as a direct and effective interface for situated 3D language reasoning, reducing the need to translate an agent's geometric state into language before reasoning.

AI USE STATEMENT

In this work, we used generative AI tools to provide feedback on research framing and experimental design, and to assist in translation and implementing methods. We have not used generative AI tools to generate synthetic datasets, formulate or prove mathematical claims, or interpret results, and the remaining required-disclosure tasks are not applicable to this work. Additionally, we used generative AI tools for literature search and discovery and polishing the writing for clarity and readability. We reviewed all AI-assisted work, including independently verifying technical claims and citations against the original sources and experimental results. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

Section 3 specifies the agent-centric transformation, point-token interface, and egocentric alignment procedure, while Section 4 describes the datasets, evaluation metrics, and experimental setup. The Appendices A– C provide alignment-data construction details, training hyperparameters, evaluation protocols, statistical tests, and complete numerical results. To facilitate reproduction and future research, we will release our training and evaluation code, processed alignment annotations, model checkpoints, and evaluation outputs upon publication.

REFERENCES

Cesar Cadena, Luca Carlone, Henry Carrillo, Yasir Latif, Davide Scaramuzza, José Neira, Ian Reid, and John J Leonard. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on robotics*, 32(6):1309–1332, 2016.

Carlos Campos, Richard Elvira, Juan J Gómez Rodríguez, José MM Montiel, and Juan D Tardós. Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam. *IEEE transactions on robotics*, 37(6):1874–1890, 2021.

Sijin Chen, Xin Chen, Chi Zhang, Mingsheng Li, Gang Yu, Hao Fei, Hongyuan Zhu, Jiayuan Fan, and Tao Chen. Ll3da: Visual interactive instruction tuning for omni-3d understanding, reasoning, and planning. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 26418–26428. IEEE, 2024a.

Yilun Chen, Shuai Yang, Haifeng Huang, Tai Wang, Runsen Xu, Ruiyuan Lyu, Dahua Lin, and Jiangmiao Pang. Grounded 3d-llm with referent tokens. *arXiv preprint arXiv:2405.10370*, 2024b.

Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *2017 IEEE conference on computer vision and pattern recognition (CVPR)*, pp. 2432–2443. IEEE, 2017.

Jiajun Deng, Tianyu He, Li Jiang, Tianyu Wang, Feras Dayoub, and Ian Reid. 3d-llava: Towards generalist 3d lmms with omni superpoint transformer. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3772–3782. IEEE, 2025.

Rao Fu, Jingyu Liu, Xilun Chen, Yixin Nie, and Wenhan Xiong. Scene-llm: Extending language model for 3d visual understanding and reasoning. *arXiv preprint arXiv:2403.11401*, 2024.

Yining Hong, Haoyu Zhen, Peihao Chen, Shuhong Zheng, Yilun Du, Zhenfang Chen, and Chuang Gan. 3d-llm: Injecting the 3d world into large language models. *Advances in Neural Information Processing Systems*, 36:20482–20494, 2023.

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

Jiangyong Huang, Silong Yong, Xiaojian Ma, Xiongkun Linghu, Puhao Li, Yan Wang, Qing Li, Song-Chun Zhu, Baoxiong Jia, and Siyuan Huang. An embodied generalist agent in 3d world. *arXiv preprint arXiv:2311.12871*, 2023.

Jiangyong Huang, Baoxiong Jia, Yan Wang, Ziyu Zhu, Xiongkun Linghu, Qing Li, Song-Chun Zhu, and Siyuan Huang. Unveiling the mist over 3d vision-language understanding: Object-centric evaluation with chain-of-analysis. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 24570–24581. IEEE, 2025.

Lucy Lin, Ayush Jain, Yifan Liu, and Katerina Fragkiadaki. Qwen-3d: A generalist 3d vision-language model for spatial understanding. In *European Conference on Computer Vision*, pp. 186–204. Springer, 2026.

Xiongkun Linghu, Jiangyong Huang, Xuesong Niu, Xiaojian Ma, Baoxiong Jia, and Siyuan Huang. Multi-modal situated reasoning in 3d scenes. *Advances in Neural Information Processing Systems*, 37:140903–140936, 2024.

Xiongkun Linghu, Jiangyong Huang, Ziyu Zhu, Baoxiong Jia, and Siyuan Huang. Scenecot: Eliciting grounded chain-of-thought reasoning in 3d scenes. In *International Conference on Learning Representations*, volume 2026, pp. 2754–2779, 2026.

Quan Liu, Weihao Xuan, Junjue Wang, Naoto Yokoya, Ling Shao, and Shijian Lu. Direction-aware 3d large multimodal models. *arXiv preprint arXiv:2602.19063*, 2026.

Xianzheng Ma, Tao Sun, Shuai Chen, Yash Bhalgat, Jindong Gu, Angel Chang, Iro Armeni, Iro Laina, Songyou Peng, and Victor Prisacariu. Do 3d large language models really understand 3d spatial relationships? In *International Conference on Learning Representations*, volume 2026, pp. 32591–32610, 2026.

Xiaojian Ma, Silong Yong, Zilong Zheng, Qing Li, Yitao Liang, Song-Chun Zhu, and Siyuan Huang. Sqa3d: Situated question answering in 3d scenes. *arXiv preprint arXiv:2210.07474*, 2022.

Yunze Man, Liang-Yan Gui, and Yu-Xiong Wang. Situational awareness matters in 3d vision language reasoning. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13678–13688. IEEE, 2024.

Guofeng Mei, Wei Lin, Luigi Riz, Yujiao Wu, Fabio Poiesi, and Yiming Wang. Perla: Perceptive 3d language assistant. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14369–14379. IEEE, 2025.

Zhangyang Qi, Zhixiong Zhang, Ye Fang, Jiaqi Wang, and Hengshuang Zhao. Gpt4scene: Understand 3d scenes from videos with vision-language models. In *International Conference on Learning Representations*, volume 2026, pp. 157277–157302, 2026.

Qwen Team. Qwen2.5: A party of foundation models, September 2024. URL `https://qwenlm.github.io/blog/qwen2.5/`.

Zehan Wang, Haifeng Huang, Yang Zhao, Ziang Zhang, and Zhou Zhao. Chat-3d: Data-efficiently tuning large language model for universal dialogue of 3d scenes. *arXiv preprint arXiv:2308.08769*, 2023.

Xiaoyang Wu, Daniel DeTone, Duncan Frost, Tianwei Shen, Chris Xie, Nan Yang, Jakob Engel, Richard Newcombe, Hengshuang Zhao, and Julian Straub. Sonata: Self-supervised learning of reliable point representations. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22193–22204. IEEE, 2025.

Runsen Xu, Xiaolong Wang, Tai Wang, Yilun Chen, Jiangmiao Pang, and Dahua Lin. Pointllm: Empowering large language models to understand point clouds. In *European Conference on Computer Vision*, pp. 131–147. Springer, 2024.

Zhihao Yuan, Yibo Peng, Jinke Ren, Yinghong Liao, Yatong Han, Chun-Mei Feng, Hengshuang Zhao, Guanbin Li, Shuguang Cui, and Zhen Li. Empowering large language models with 3d situation awareness. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19435–19445. IEEE, 2025.

Yue Zhang, Zhiyang Xu, Ying Shen, Parisa Kordjamshidi, and Lifu Huang. Spartun3d: Situated spatial understanding of 3d world in large language models. *arXiv preprint arXiv:2410.03878*, 2024.

Yujia Zhang, Xiaoyang Wu, Yixing Lao, Chengyao Wang, Zhuotao Tian, Naiyan Wang, and Hengshuang Zhao. Concerto: Joint 2d-3d self-supervised learning emerges spatial representations. *Advances in Neural Information Processing Systems*, 38:69498–69522, 2026.

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.

Hongyan Zhi, Peihao Chen, Junyan Li, Shuailei Ma, Xinyu Sun, Tianhang Xiang, Yinjie Lei, Mingkui Tan, and Chuang Gan. Lscenellm: Enhancing large 3d scene understanding using adaptive visual preferences. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3761–3771. IEEE, 2025.

Chenming Zhu, Tai Wang, Wenwei Zhang, Jiangmiao Pang, and Xihui Liu. Llava-3d: A simple yet effective pathway to empowering lmms with 3d-awareness. *arXiv preprint arXiv:2409.18125*, 2024.

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 POSE-SHUFFLE EVALUATION AND SIGNIFICANCE

Pose-shuffle evaluation. For each SQA3D test scene, we place its distinct annotated agent poses in a deterministic random cycle and replace each pose with its successor. Questions sharing an original pose receive the same donor pose, and no pose maps to itself. The donor mapping is shared across methods; the scene, question, answer, and situation description (when present) remain fixed. We compare correct- and shuffled-pose inputs using the same checkpoint for each method.

Table 5 reports paired pose-shuffle significance tests with and without situation descriptions. We compute 95% confidence intervals from per-question paired accuracy differences using a normal approximation, and report two-sided exact McNemar p-values. For LEO and PoseAlign, pose shuffling causes only negligible changes in EM and EM-R under both settings; all 95% confidence intervals include zero and none of the differences are statistically significant ($p \geq .10$). In contrast, EgoAlign exhibits large and highly significant drops in both settings. With situation descriptions, shuffling the pose reduces EM/EM-R by 9.12/9.58 points, while removing situation descriptions increases the drops to 10.88/11.11 points; all corresponding confidence intervals are far from zero ($p < 10^{-39}$). These results confirm that EgoAlign is strongly dependent on the supplied numerical pose, whereas the predictions of LEO and PoseAlign remain largely pose-insensitive regardless of whether situation text is available.

Table 5: Paired SQA3D significance analysis under pose shuffling after fine-tuning with (+sit.) or without (−sit.) situation descriptions. LEO and PoseAlign are fine-tuned using their official hyper-parameters with officially released checkpoints.

Setting	Method	Metric	Correct	Shuffled	Drop	95% CI	p
+sit.	LEO	EM	49.73	49.82	−0.09	$[-0.54, 0.37]$	.80
		EM-R	52.29	52.29	0.00	$[-0.45, 0.45]$	1.00
	PoseAlign	EM	54.22	54.39	−0.17	$[-1.17, 0.82]$	.78
		EM-R	56.89	56.92	−0.03	$[-1.05, 0.99]$	1.00
	EgoAlign	EM	61.21	52.09	9.12	$[7.76, 10.49]$	$< 10^{-39}$
		EM-R	64.34	54.76	9.58	$[8.21, 10.97]$	$< 10^{-41}$
−sit.	LEO	EM	48.28	48.11	0.17	$[-0.43, 0.77]$	.64
		EM-R	50.67	50.53	0.14	$[-0.48, 0.76]$	.72
	PoseAlign	EM	52.00	51.55	0.45	$[-0.05, 0.96]$	.10
		EM-R	54.50	54.08	0.43	$[-0.09, 0.94]$	.13
	EgoAlign	EM	60.39	49.50	10.88	$[9.35, 12.41]$	$< 10^{-42}$
		EM-R	63.23	52.12	11.11	$[9.56, 12.66]$	$< 10^{-43}$

B TRAINING AND IMPLEMENTATION DETAILS

Alignment data. The SQA3D training split provides 518 ScanNet scenes and 4,262 eligible agent positions; scene-disjoint validation uses 65 scenes and 493 positions (Ma et al., 2022; Dai et al., 2017). There is no overlap between scenes used in alignment training and downstream evaluation. For each task, we sample 18,750 complete four-heading groups for training and 250 for validation. ScanNet instance annotations determine the targets but are not model inputs; each example retains the complete scene point cloud. Category uses 17 object classes after excluding structural surfaces and the catch-all *other furniture* class. For direction and horizontal relation labels, we take the angle of the target center relative to the agent (or the reference-object center) in the agent-centric xy plane and assign it to the nearest of eight directions at 45° intervals. We assign above or below only when the upper object's center lies within the lower object's horizontal bounding box and above its top. Figure 9 reports the resulting answer distributions.

Optimization. We initialize the point tower from the official Concerto Large checkpoint and use independent two-layer MLPs for the semantic, coordinate, and color projections. LoRA adapters have rank 16, $\alpha = 32$, and dropout 0.05, and are applied to all attention and MLP projections of the

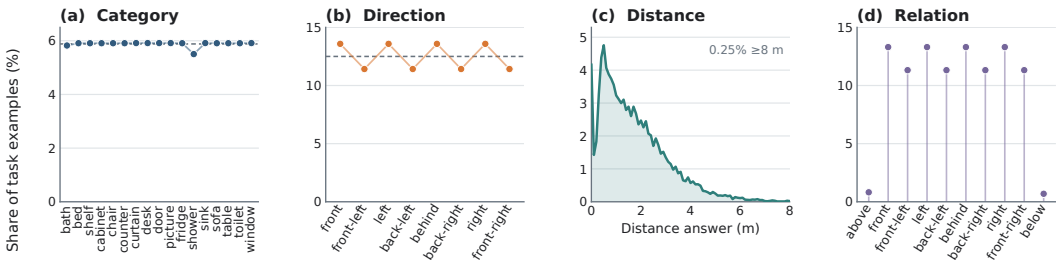


Figure 9: Answer distributions in the alignment set (75K examples per task). Dashed lines denote uniform category and direction distributions; distance retains its 0.1 m precision.

language model. Alignment runs for one 300K-example epoch with a global batch size of 8, AdamW, a peak learning rate of 5×10^{-5}, 3% linear warmup, cosine decay, BF16, and seed 42.

Point-cloud preprocessing. We voxelize each scene at 2.5 cm resolution, retaining one point per occupied voxel. The point encoder uses voxel indices for sparse operations while retaining the selected points' continuous agent-centric XYZ coordinates. Before the coordinate projector, token-pooled XYZ is divided elementwise by $(16, 16, 4)$ m, placing horizontal and vertical offsets on comparable numerical scales while retaining their agent-relative signs.

Task-specific fine-tuning. SQA3D uses all 26,623 training questions for three epochs. MSQA uses 234,639 training examples for one epoch. Both use a global batch size of 4 and the alignment-stage optimizer, learning rate, warmup ratio, and precision. Real-3DQA is directly evaluated using the SQA3D-finetuned checkpoints.

C Additional Results

C.1 Alignment Dynamics of Component Ablations

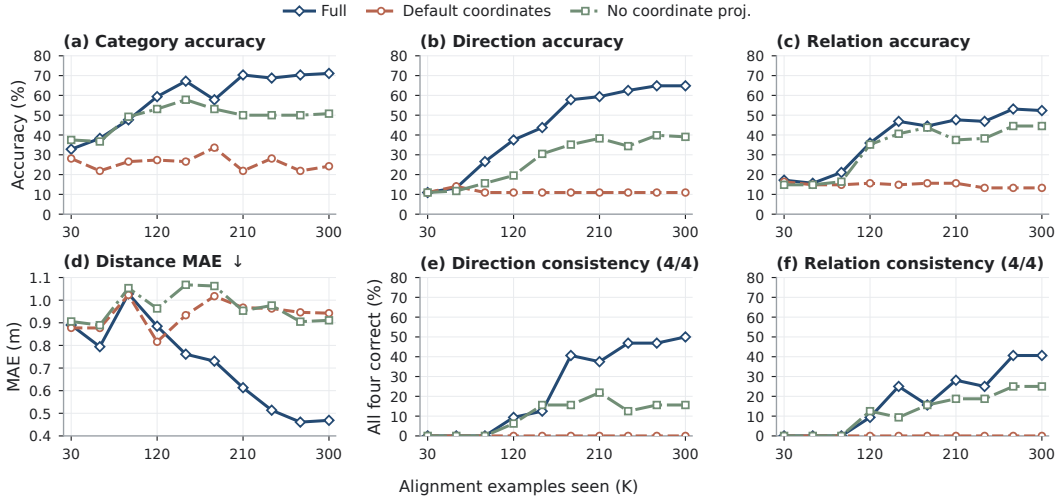


Figure 10: Alignment dynamics for the Qwen2.5-1.5B component ablations.

Figure 10 expands the component ablations in Table 3 by tracking alignment performance throughout training. In addition to per-example accuracy, we report 4/4 consistency for the direction and relation tasks, defined as the fraction of four-heading counterfactual groups for which all four variants are answered correctly. The full model consistently outperforms both ablations across the four alignment tasks and the consistency metrics. In particular, it reaches substantially higher 4/4 consistency, showing that EgoAlign learns to update its predictions coherently as the numerical heading changes. Using the default scene coordinates largely prevents the model from learning heading-dependent

direction and relation predictions, while removing the explicit coordinate projector yields intermediate performance. These results further show that transforming the scene into the agent frame and explicitly exposing metric coordinates to the LLM are complementary.

C.2 Complete model-scale results

Table 6: Full SQA3D and Real-3DQA results across model scales (%). Type columns report VRS within each question type; darker shading indicates higher performance.

		SQA3D		Real-3DQA						
Size	Input	EM	EM-R	EM	EM-R	VRS	Dist.	Dir.	Count.	Exist.
1.5B	+sit.	58.0	60.6	30.4	35.8	16.1	12.9	26.5	21.7	61.5
	−sit.	56.8	59.7	30.2	35.3	16.9	14.0	25.4	19.2	67.3
3B	+sit.	58.3	61.1	30.9	36.2	16.2	13.1	27.2	22.5	46.2
	−sit.	58.1	60.6	28.6	33.5	16.7	14.1	25.0	20.8	53.8
7B	+sit.	61.2	64.3	33.3	38.5	18.6	16.4	25.9	18.3	57.7
	−sit.	60.4	63.2	33.7	38.6	19.1	16.5	27.2	22.5	57.7

Table 7: Full MSQA results across model scales (%). Darker shading indicates higher performance within each column.

		MSQA						
Size	Input	Count.	Exist.	Attr.	Spatial	Nav.	Other	Overall
1.5B	+sit.	38.2	91.1	48.7	49.0	57.0	82.6	56.9
	−sit.	37.6	89.8	51.3	53.4	60.1	84.1	58.9
3B	+sit.	39.6	93.9	52.0	51.2	57.9	82.0	58.7
	−sit.	40.5	91.6	54.5	54.6	62.7	83.7	60.9
7B	+sit.	36.7	91.7	51.7	51.9	59.5	84.1	58.5
	−sit.	37.0	93.0	54.7	54.0	61.2	83.1	60.0

Tables 6 and 7 provide the complete SQA3D, Real-3DQA, and MSQA results underlying the model-scale analysis in the main paper. Scaling generally improves SQA3D and Real-3DQA, with the 7B model obtaining the strongest overall results, whereas MSQA peaks at 3B. Across model sizes, removing situation descriptions remains competitive on SQA3D and consistently improves the overall MSQA score. On Real-3DQA, the no-situation variants also achieve comparable or higher viewpoint-robustness scores, indicating that numerical pose remains effective across model scales.

C.3 Complete yaw-noise results

Table 8 reports the exact values corresponding to Figure 7. Performance generally decreases as the numerical heading becomes less accurate across all model scales and both input settings. A 90° yaw error reduces SQA3D EM by 8.9–10.6 points relative to the clean pose, further demonstrating a consistent dependence on the supplied numerical reference frame.

C.4 Additional qualitative examples

Figure 11 provides additional four-heading comparisons on Real-3DQA. Within each row, the scene, agent position, and question remain fixed while only the heading changes. EgoAlign generally adapts its predictions to the changing reference frame, whereas PoseAlign more frequently repeats answers across incompatible headings. EgoAlign is not perfect and still makes errors on some fine-grained closest-object cases, but overall the examples are consistent with the quantitative pose-sensitivity results in the main paper.

Table 8: SQA3D EM (%) under numerical yaw perturbations. Zero denotes the unperturbed pose; column headings give absolute yaw error in degrees.

Size	Input	0°	15°	30°	45°	60°	75°	90°
1.5B	+sit.	58.0	57.4	56.3	54.3	51.8	49.8	48.3
	−sit.	56.8	56.2	54.8	53.4	51.5	49.1	47.1
3B	+sit.	58.3	57.7	57.0	55.4	53.0	50.6	48.8
	−sit.	58.1	58.1	57.3	55.4	53.7	51.5	49.2
7B	+sit.	61.2	60.3	59.7	57.5	55.3	53.3	51.3
	−sit.	60.4	59.9	58.9	56.7	54.2	51.2	49.8

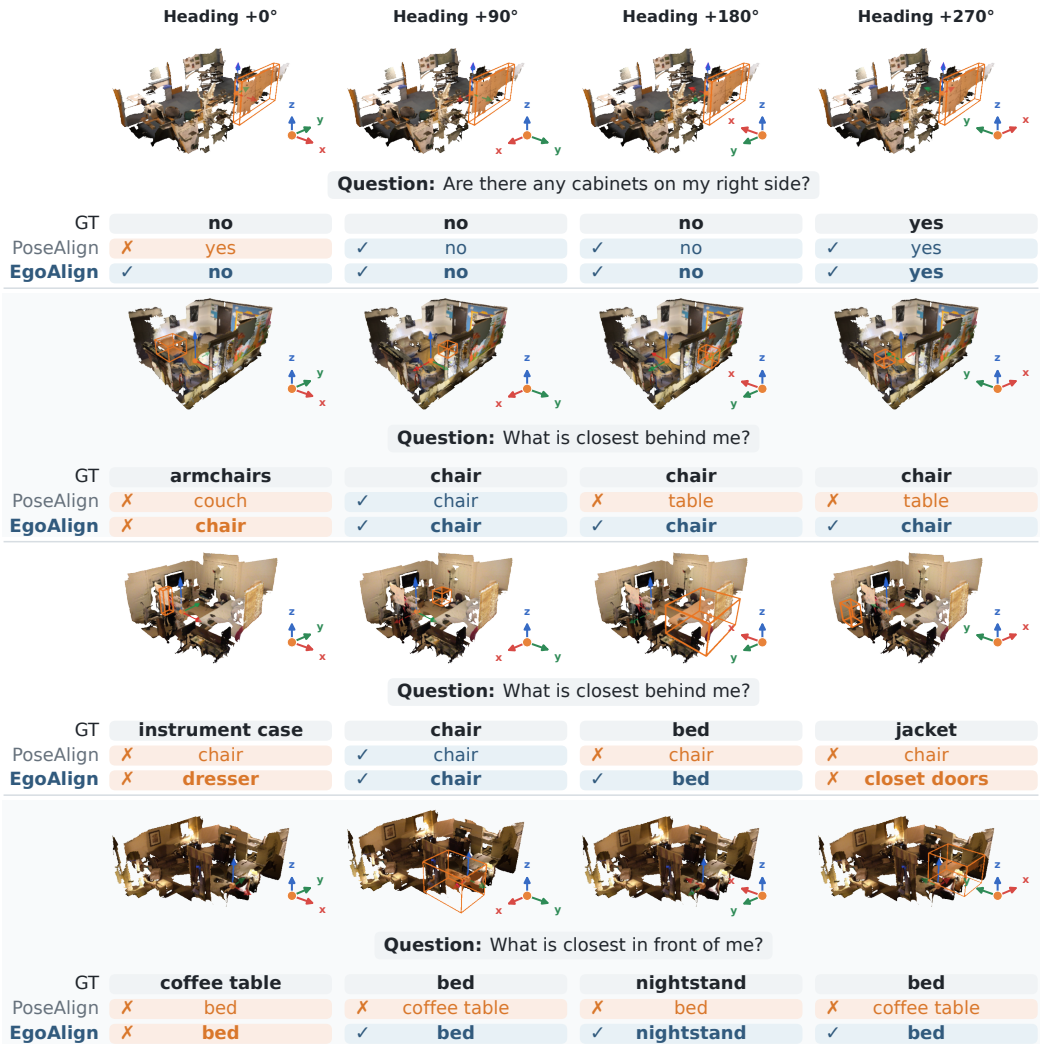


Figure 11: Additional four-heading Real-3DQA examples. Within each row, only the agent heading changes. Boxes mark matching ScanNet instances in the queried direction.