An Embodied Generalist Agent in 3D World

Jiangyong Huang [*1,2] **Silong Yong** [*1,3] **Xiaojian Ma** [*,†1] **Xiongkun Linghu** [*1] **Puhao Li** [1,3]
Yan Wang [1] **Qing Li** [1] **Song-Chun Zhu** [1,2,3] **Baoxiong Jia** [1] **Siyuan Huang** [†1]

Abstract

Leveraging massive knowledge from large language models (LLMs), recent machine learning models show notable successes in general-purpose task solving in diverse domains such as computer vision and robotics. However, several significant challenges remain: (i) most of these models rely on 2D images yet exhibit a limited capacity for 3D input; (ii) these models rarely explore the tasks inherently defined in 3D world, *e.g.*, 3D grounding, embodied reasoning and acting. We argue these limitations significantly hinder current models from performing real-world tasks and approaching general intelligence. To this end, we introduce LEO, an embodied multi-modal generalist agent that excels in perceiving, grounding, reasoning, planning, and acting in the 3D world. LEO is trained with a unified task interface, model architecture, and objective in two stages: (i) 3D vision-language (VL) alignment and (ii) 3D vision-language-action (VLA) instruction tuning. We collect large-scale datasets comprising diverse object-level and scene-level tasks, which require considerable understanding of and interaction with the 3D world. Moreover, we meticulously design an LLM-assisted pipeline to produce high-quality 3D VL data. Through extensive experiments, we demonstrate LEO's remarkable proficiency across a wide spectrum of tasks, including 3D captioning, question answering, embodied reasoning, navigation and manipulation. Our ablative studies and scaling analyses further provide valuable insights for developing future embodied generalist agents. Code and data are available on project page.

[*]Equal contribution [†]Research lead [1]State Key Laboratory of General Artificial Intelligence, Beijing Institute for General Artificial Intelligence (BIGAI) [2]Peking University [3]Tsinghua University.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

1. Introduction

Building one generalist model that can handle comprehensive tasks like humans has been a long-existing pursuit in artificial intelligence and neuroscience (Lake et al., 2015; 2017; Zhu et al., 2020; Mountcastle, 1979; Schmidhuber, 2018; Huang et al., 2022a). Recent advances in LLMs (Brown et al., 2020) and "foundation models" (Bommasani et al., 2021) emerge as a promising paradigm in building such generalist models in natural language processing (OpenAI, 2022; 2023), computer vision (Kirillov et al., 2023; Alayrac et al., 2022), and robotics (Brohan et al., 2022; 2023; Reed et al., 2022; Driess et al., 2023; Li et al., 2023c). The keys to the success of this paradigm lie in large-scale internet-level datasets from numerous tasks and domains, as well as scalable Transformer architectures (Vaswani et al., 2017) that can absorb generalizable and task-agnostic knowledge from the data. Nonetheless, existing generalist models primarily thrive within 2D domains, lacking comprehension of the 3D physical environment that envelops human-level intelligence. This limitation stands as an obstacle that prevents current models from solving real-world tasks and approaching general intelligence. Therefore, we ask a fundamental question: *how to equip the generalist agent with a comprehensive understanding of and the ability to interact with the real 3D world*?

The development of such generalist agents encounters three primary challenges: the lack of suitable datasets, unified models, and effective learning strategies. Despite substantial progress in scaling up image-text models (Tsimpoukelli et al., 2021; Alayrac et al., 2022) and the curation of corresponding datasets (Radford et al., 2021; Schuhmann et al., 2022), advancement in 3D scene-level understanding has significantly lagged behind. This is largely attributed to the limited scale and manual labeling of 3D datasets (Dai et al., 2017; Wald et al., 2019; Chen et al., 2020), given the higher cost associated with collecting 3D data compared to 2D data. Furthermore, large-scale unified pretraining and efficient finetuning are under-explored by previous 3D VL models, which are often designed with strong priors (Zhao et al., 2021; Chen et al., 2022). Notably, recent works (Zhu et al., 2023c; Hong et al., 2023) utilize multi-modal Transformer together with synthetic data to enhance the model's capability in grounded 3D scene understanding. Nevertheless,

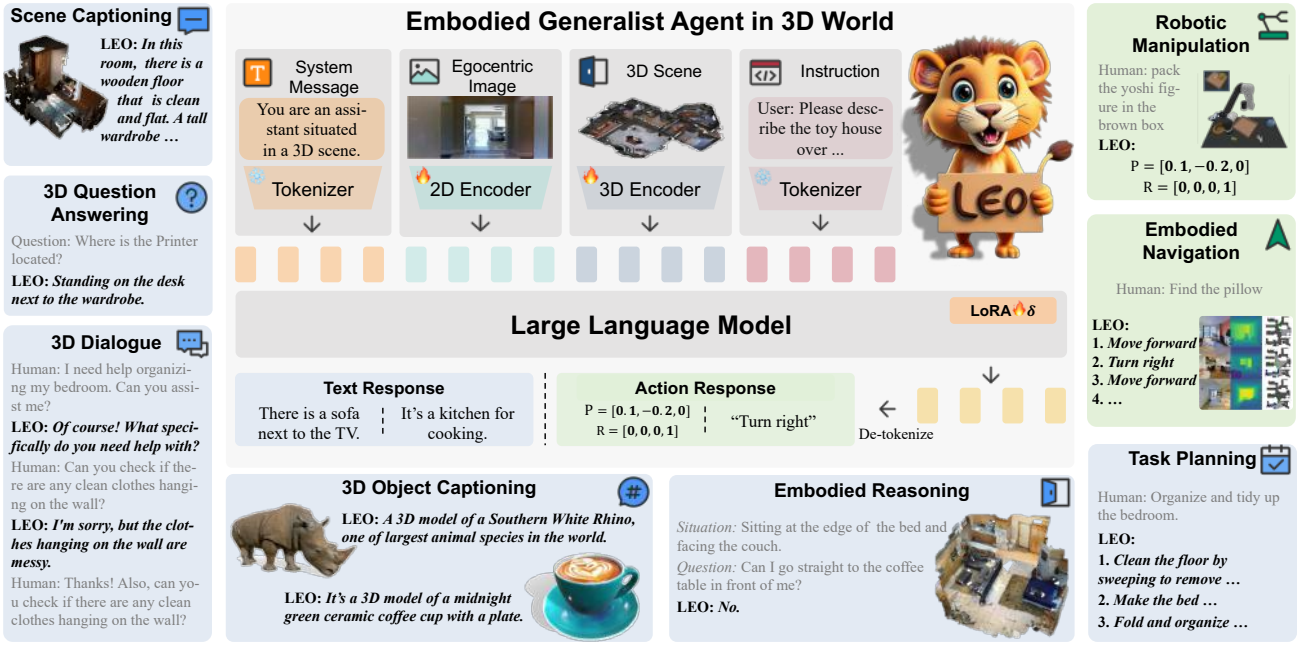


Figure 1: **The proposed embodied generalist agent LEO**. It takes egocentric 2D images, 3D point clouds, and texts as input and formulates comprehensive 3D tasks as autoregressive sequence prediction. By instruction-tuning LEO, it extends the capability of LLMs to multi-modal vision-language-action tasks with a unified model.

they fall short in embodied tasks, *e.g.*, acting within 3D environments. Additionally, there are significant yet rarely explored problems, *e.g.*, the potential of VLA learning and efficient adaptation of LLMs for 3D tasks.

In this work, we introduce the generalist agent LEO, which is generically embodied, multi-modal, and general-purpose. It can take egocentric 2D images, 3D point clouds, and texts as task input and handle comprehensive tasks within the 3D environment. As shown in Fig. 1, LEO *exhibits the capability of perceiving, grounding, reasoning, planning, and acting with a unified task interface, model architecture, and objective.* LEO perceives through an egocentric 2D image encoder for the embodied view and an object-centric 3D point cloud encoder for the third-person global view. Such perception modules can be flexibly adapted to various embodied environments and enhance 3D reasoning. The encoded visual tokens are interleaved with text tokens to form a unified multi-modal task sequence, which further serves as the input to a decoder-only LLM. Equipped with a vocabulary containing both text and action tokens, the LLM can generate responses to various tasks simultaneously. Consequently, all the tasks are formulated as sequence prediction, thereby accommodating a unified training objective.

Following prior experiences (Liu et al., 2023b), we adopt a two-stage learning scheme, *i.e.*, 3D VL alignment and 3D VLA instruction tuning. We accordingly collect large-scale comprehensive datasets LEO-align and LEO-instruct, which comprise diverse object-level and scene-level tasks. Notably, we meticulously design an LLM-assisted pipeline to generate high-quality 3D VL data, wherein we propose to prompt LLMs (OpenAI, 2022) with scene graphs and Object-centric Chain-of-Thought (O-CoT) method. To further enhance quality control, we devise a series of refinement procedures via regular expression matching and scene graph retrieval. We demonstrate our approach largely enriches the data scale and diversity, meanwhile mitigating hallucination in LLM-generated data.

We quantitatively evaluate and ablate LEO on diverse 3D tasks, including 3D captioning (Chen et al., 2021), 3D question answering (Azuma et al., 2022), situated question answering (Ma et al., 2023), embodied navigation (Ramrakhya et al., 2022), and robotic manipulation (Shridhar et al., 2021). The results indicate (i) through task-agnostic instruction tuning with a unified model, LEO achieves state-of-the-art performances on most tasks, particularly surpassing previous task-specific models; (ii) LEO shows proficiency in scene-grounded dialogue and planning, capable of generating flexible and coherent responses; (iii) LEO achieves comparable performances to state-of-the-art task-specific models on navigation and manipulation tasks, and exhibits remarkable generalization ability; (iv) LEO's strong performances stem from both data and model aspects, including the alignment stage, data diversity, generalist-style instruction tuning, and object-centric representation; (v) LEO manifests the scaling law that echoes prior findings (Kaplan et al., 2020; Reed et al., 2022; OpenAI, 2023). We also present qualitative results to illustrate LEO's versatility and proficiency in grounded 3D scene understanding.

In summary, our main contributions are as follows: (i) we propose LEO, the first embodied generalist agent capable of following human instructions to perceive, ground, reason, plan, and act in the 3D world; (ii) we propose a simple yet effective framework that connects object-centric 3D representation and LLM to efficiently bridge the gap between vision, language, and embodied action; (iii) we collect large-scale comprehensive datasets for our two-stage generalist training scheme, and particularly propose an LLM-assisted pipeline for the generation of high-quality 3D VL data; (iv) we conduct extensive experiments to demonstrate LEO's proficiency across various tasks, and present in-depth analyses to reveal valuable insights; (v) we release the data, code, and model weights to endow the future research in embodied generalist agents.

2. Model

The leading design principles of LEO are two-fold: 1) It should handle the multi-modal input of egocentric 2D, global 3D, and textual instruction, and the output of textual response as well as embodied action commands in a unified architecture; 2) It should leverage pre-trained large language models (LLMs) as a powerful prior for the downstream tasks. We therefore convert all data of different modalities into a sequence of tokens, illustrated below:

$$\underbrace{\texttt{You are...}}_{\text{system message}} \underbrace{s_{\text{2D}}^{(1)}, ..., s_{\text{2D}}^{(M)}}_{\substack{\text{2D image tokens} \\ \text{(optional)}}} \underbrace{s_{\text{3D}}^{(1)}, ..., s_{\text{3D}}^{(N)}}_{\substack{\text{object-centric} \\ \text{3D tokens}}}, \quad \underbrace{\texttt{USER:... ASSISTANT:}}_{\text{instruction}} \underbrace{s_{\text{res}}^{(1)}, ...s_{\text{res}}^{(T)}}_{\text{response}}. \tag{1}$$

With this representation, we formulate the learning of LEO as GPT-style autoregressive language modeling (Brown et al., 2020) given the *prefix* (from *system message* to *instruction*), *i.e.* prefix language modeling (Raffel et al., 2020). Therefore, a pretrained LLM can be used to process such sequences. Next, we will detail the tokenization of multi-modal data, model architecture, training loss, and inference settings. An overview of our model can be found in Fig. 1.

2.1. Tokenization

We follow prior practices in 2D VLM (Liu et al., 2023b; Alayrac et al., 2022) and 3D VLM (Zhu et al., 2023c) to tokenize the multi-modal data in LEO. We use SentencePiece tokenizer (Kudo & Richardson, 2018) to encode text with 32k subwords; 2D image tokens for egocentric 2D images; and object-centric 3D tokens extracted over Mask3D-based (Schult et al., 2022) object proposals for 3D point cloud inputs. For embodied action commands, continuous actions (*e.g.* in manipulation) are discretized (details in Appendix D.3) to join the discrete actions (*e.g.* navigation) and form a unified discrete action space. We follow (Brohan et al., 2023) to map these discrete actions to the least used tokens in SentencePiece. After tokenization, all tokens are ordered into the format in (1).

2.2. Token Embedding & LLM

We apply several token embedding functions to process the tokens in the sequence before sending them to the LLM. The LLM will then align these tokens of different modalities, and produce the response. Most of the responses are text and can be decoded directly. For responses that include embodied actions, we will map the reserved SentencePiece text tokens back to action commands.

Text & 2D token embedding. For text tokens (including embodied actions that have been mapped to the reserved text tokens), an embedding look-up table is used to map them into vectors. While the egocentric 2D image is encoded by a pretrained OpenCLIP ConvNext (Liu et al., 2022) for obtaining image token embeddings. We apply MLP adapters to match the dimensions of all token embeddings.

Object-centric 3D token embedding. Each 3D object token (*i.e.*, the point cloud of a 3D object) is first encoded by a pretrained point cloud encoder (*e.g.*, PointNet++ (Qi et al., 2017)). We then adopt the Spatial Transformer introduced in (Chen et al., 2022) to further process the point cloud embedding of all objects into object-centric 3D token embeddings. In a nutshell, Spatial Transformer biases the standard attention score with relative position and size for capturing 3D relations between objects. Due to space limit, the readers are referred to (Chen et al., 2022) and Appendix D.2 for more details.

Pretrained LLM. We choose Vicuna-7B (Chiang et al., 2023) to process the token sequence. In order to tackle the challenging alignment and grounding problem of multi-modal tokens (2D, 3D, text, embodied action) while preserving the LLM pretrained knowledge, we employ LoRA (Hu et al., 2022) to introduce additional tunable parameters to the frozen pretrained LLM.

2.3. Training & Inference

We formulate the learning objective of LEO following (Brown et al., 2020; Raffel et al., 2020) in a prefix language modeling fashion. For a batch $\mathcal{B}$ of token sequence s, we optimize LEO via:

$$\mathcal{L}(\theta, \mathcal{B}) = -\sum_{b=1}^{|\mathcal{B}|} \sum_{t=1}^{T} \log p_\theta(s_{\text{res}}^{(b,t)} | s_{\text{res}}^{(b,<t)}, s_{\text{prefix}}^{(b)}), \tag{2}$$

where s_{prefix} denotes the prefix tokens (from *system message* to *instruction*) in (1). During training, we freeze the pretrained 3D point cloud encoder and the LLM and fine-tune the 2D image encoder, the Spatial Transformer, and the LoRA parameters. In total, LEO has ~7B parameters and ~142M of them will be tuned. During inference, we

Table 1: **Datasets statistics**. We illustrate key statistics of datasets for 3D VL alignment (LEO-align) and 3D VLA instruction tuning (LEO-instruct). *res.* (response) denotes tokens to be predicted, while *prefix* denotes those in the context.

Dataset	Task	2D input	3D assets	#data	#token (*res.*)	#token (*prefix*+*res.*)
LEO-align	object captioning	✗	Objaverse	660K	10M	27M
	object referring	✗	ScanNet + 3RScan	354K	15M	39M
	scene captioning	✗	3RScan	20K	3.3M	4.4M
LEO-instruct	3D captioning	✗	ScanNet	37K	821K	3M
	3D QA	✗	ScanNet + 3RScan	83K	177K	4M
	3D dialogue	✗	3RScan	11K	1.1M	8.3M
	task planning	✗	3RScan	14K	1.9M	2.7M
	navigation	✓	MP3D	60K	11.4M	272M
	manipulation	✓	CLIPort	300K	7.2M	734M

Table 2: Answer accuracy of LLM-generated data on three types of questions.

	Counting	Existence	Non-existence
3D-LLM	56.5	96.8	40.0
Ours	57.4	91.3	27.4
+ O-CoT	78.0	93.4	30.5
+ refinement	100.0	100.0	100.0

Table 3: The amount of examined data in Tab. 2. 3D-LLM data (Hong et al., 2023) is much less since we can only access a subset.

	Counting	Existence	Non-existence
3D-LLM	434	95	10
Ours	2666	6766	3314

use beam search to generate textual responses. For tasks that require action commands, we map the textual outputs to action commands as discussed in Sec. 2.1. More details on the model and training can be found in Appendix D.

3. Datasets

Since LEO is a generalist agent that receives multi-modal inputs and follows instructions, we adopt the two-stage training proposed by Liu et al. (2023b) and split the data into two sets: (i) LEO-align (Sec. 3.1) that focuses on **3D vision-language (VL) alignment** to bridge the gap between 3D scene representation and natural language; and (ii) LEO-instruct (Sec. 3.2) that targets at **3D VLA instruction tuning** to endow LEO with various capabilities. The statistics and examples of these datasets can be found in Tab. 1 and Appendix C, respectively. Due to the data scarcity, we adopt LLMs to facilitate the data generation process and outline the details in Sec. 3.3.

3.1. LEO-align: 3D Vision-Language Alignment

In LEO-align, we focus on 3D VL alignment. Similar to BLIP-2 (Li et al., 2023d), we train LEO to generate captions given various 3D inputs. Specifically, we collect three types of 3D captioning data: 1) **object-level captions**, where we align 3D individual objects with their descriptions (Luo et al., 2023); 2) **object-in-the-scene captions**, where the goal is to generate the referring expressions of objects in a 3D scene context (Achlioptas et al., 2020; Zhu et al., 2023c); and 3) **scene-level captions**, which focuses on depicting global 3D scene using natural language. Due to the space limit, we defer details including data source and components to Appendix B.1.

3.2. LEO-instruct: Instruction Following in 3D world

In LEO-instruct, LEO will be tuned to follow instructions and accomplish various 3D VLA tasks. We curate a comprehensive set of tasks that covers a broad spectrum from grounded scene understanding and reasoning (Chen et al., 2021; Ma et al., 2023), to dialogue, planning, and embodied acting (Savva et al., 2019; Shridhar et al., 2021). Specifically, we introduce 1) **3D captioning and question answering** – given 3D scene input, the agent needs to generate a natural language response to describe the scene or answer questions; 2) **3D dialogue and task planning**, where the agent is expected to generate flexible and coherent responses to complex instructions with respect to the given 3D scene, and 3) **navigation and manipulation**, which require the agent to accomplish a variety of embodied acting tasks in the 3D scene. We defer details to Appendix B.2.

3.3. LLM-assisted 3D-language Data Generation

As mentioned above, at the core of producing a large proportion of LEO-align and LEO-instruct is the assistance of LLMs. We now detail the key techniques of prompting LLMs (*i.e.*, ChatGPT) to generate 3D-text paired data. An overview can be found in Fig. 2.

Scene-graph-based prompting. Our data generation pipeline starts with 3D scene graphs from 3DSSG (Wu et al., 2021), which provide scene contexts for prompting. Compared to counterparts that utilize object boxes (Yin et al., 2023; Hong et al., 2023; Wang et al., 2023e), it offers both rich object attributes and accurate spatial relation information among objects, allowing LLMs to generate data with high-quality 3D details (comparisons in Appendix B.8). Next, we manually design some examples as seed tasks (Liu et al., 2023b), including scene and object captioning, QA, dialogue, and planning, and ask LLM to produce more tasks as well as the responses. Details for designing the seed tasks can be found in Appendix B.3.

Object-centric CoT. To further combat the **hallucination** of LLMs (Bang et al., 2023) in open-ended generation as in our pipeline, we propose the object-centric chain of thought (O-CoT) prompting that requires the LLM to explicitly provide the label and ID of object candidates as thoughts during text generation. We also utilize subgraph sampling to further enhance the diversity of 3D scene graphs (see details in

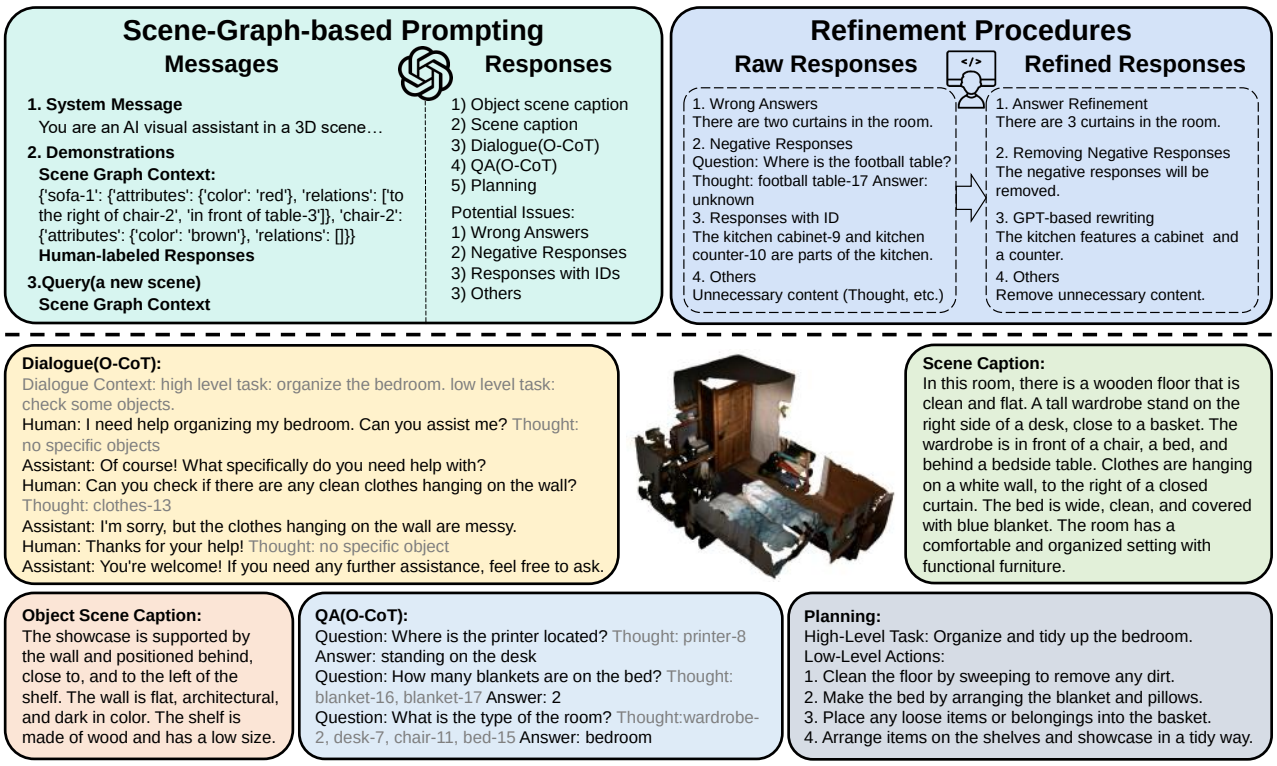


Figure 2: **Our proposed LLM-assisted 3D-language data generation pipeline and data examples.**. (Top-left) Messages with 3D scene graphs, including object attributes and relations in a phrasal form, used for providing scene context when prompting LLM. (Top-right) The human-defined refinement procedures were conducted over raw LLM responses to improve data quality. (Bottom) Examples of LLM-assisted generation in LEO-align and LEO-instruct. Thoughts, colored in gray, will be removed after refinements.

Appendix B.7). We provide examples of O-CoT in Fig. 2.

Refinement procedures. Upon the scene graph and O-CoT prompting, we introduce refinement as an additional safeguard to the quality and reliability of our generated data. Specifically, we send raw LLM responses to several human-defined filters based on the 3D scene graphs: negative responses (*e.g.*, lacking the necessary information to answer) will be removed; unnatural narratives will be rewritten, *etc*. Further, we detect text that involves logical reasoning (*e.g.*, counting) or hallucination, and manually fix the wrong responses according to the ground truth provided by scene graphs. We provide illustrative examples in Fig. 2 and Appendix B.6, and quantitative analysis on the impact of data refinement procedures in Appendix I.1.

Assess the quality of generated data. In addition to data examples, we propose to assess the quality of generated data quantitatively. We focus on the LLM-produced question-answer pairs about objects (questions starting with *How many/Is there* and ending with *in the room/bedroom/kitchen/living room/bathroom*). We first divide these pairs into three categories: *counting*, *existence*, and *non-existence*, which examines the number of certain objects/whether an object exists/whether an object does not exist in the scene, respectively. We manually check if the answers in these pairs are correct, and report the overall accuracy. Results in Tab. 2 demonstrate that our proposed scene-graph-based prompting, O-CoT prompting and refinement bring consistent improvement to data quality and the complete data generation pipeline outperforms a recent counterpart (3D-LLM). We also demonstrate how we help address the **grammatical errors** compared to counterparts in Appendix B.9. Finally, we provide the data distribution in Appendix B.10 to illustrate the **diversity** of our generated data.

4. Capabilities and Analyses

We demonstrate LEO's capabilities by a comprehensive evaluation on the full spectrum of embodied 3D tasks encompassing perceiving, grounding, reasoning, planning, and acting. In Sec. 4.1, we present quantitative comparisons between LEO and state-of-the-art models on various 3D VL tasks, underscoring LEO's proficiency in 3D VL understanding and reasoning. In Sec. 4.2, we highlight LEO's strength in scene-grounded dialogue and task planning. In Sec. 4.3, we extend LEO to embodied acting tasks wherein LEO exhibits remarkable versatility. In Sec. 4.4, we conduct ablative studies to reveal more insights into LEO, including data and model aspects. In Sec. 4.5, we probe the scaling effect and manifest the potential for further development.

Table 4: **Quantitative comparison with state-of-the-art models on 3D VL understanding and embodied reasoning tasks**. "C" stands for "CIDEr", "B-4" for "BLEU-4", "M" for "METEOR", "R" for "ROUGE", "Sim" for sentence similarity, and "EM@1" for top-1 exact match. The n-gram metrics for Scan2Cap are governed by IoU@0.5. † indicates answering questions via prompting GPT-3 with the generated scene caption. Gray indicates evaluation results with refined exact-match protocol.

	Scan2Cap (val)					ScanQA (val)					SQA3D (test)
	C	B-4	M	R	Sim	C	B-4	M	R	EM@1	EM@1
Task-specific models											
Scan2Cap	35.2	22.4	21.4	43.5	-	-	-	-	-	-	41.0†
3DJCG	47.7	31.5	24.3	51.8	-	-	-	-	-	-	-
Vote2Cap-DETR	61.8	34.5	26.2	54.4	-	-	-	-	-	-	-
ScanRefer+MCAN	-	-	-	-	-	55.4	7.9	11.5	30.0	18.6	-
ClipBERT	-	-	-	-	-	-	-	-	-	-	43.3
ScanQA	-	-	-	-	-	64.9	10.1	13.1	33.3	21.1	47.2
Task-specific fine-tuned											
3D-VisTA	66.9	34.0	27.1	54.3	53.8	69.6	10.4	13.9	35.7	22.4	48.5
3D-LLM (FlanT5)	-	-	-	-	-	69.4	12.0	14.5	35.7	20.5	-
LEO	**72.4**	**38.2**	**27.9**	**58.1**	**55.3**	**101.4**	**13.2**	**20.0**	**49.2**	**24.5** (47.6)	**50.0** (52.4)

Table 5: **Results on robot manipulation**. seen indicates in-domain tasks. unseen marks OOD tasks with novel colors or objects.

	separating-piles		packing-google-objects-seq		put-blocks-in-bowls	
	seen	unseen	seen	unseen	seen	unseen
CLIP-only	90.2	71.0	95.8	57.8	97.7	44.5
CLIPort (single)	98.0	**75.2**	**96.2**	71.9	**100**	25.0
CLIPort (multi)	89.0	62.8	84.4	70.3	**100**	**45.8**
LEO	**98.8**	**75.2**	76.6	**79.8**	86.2	35.2

Table 6: **Results on object navigation.** † indicates zero-shot evaluation.

	MP3D-val		HM3D-val	
	Success(↑)	SPL(↑)	Success(↑)	SPL(↑)
Habitat-web (shortest)	4.4	2.2	-	-
Habitat-web (demo)	**35.4**	10.2	-	-
ZSON	15.3†	4.8†	**25.5**	12.6
LEO	23.1	**15.2**	23.1†	**19.1**†

4.1. 3D Vision-Language Understanding and Reasoning

Overview. Understanding and reasoning about object attributes, object relations, and other facets of 3D scenes from an agent's egocentric perspective is a fundamental capability of an embodied generalist agent in the 3D world. We investigate *how well can* LEO *perform 3D VL understanding and embodied reasoning tasks, especially when being compared against task-specific models and existing generalist agents*. Specifically, we consider three renowned 3D tasks: 3D captioning on Scan2Cap (Chen et al., 2021), 3D QA on ScanQA (Azuma et al., 2022), and 3D embodied reasoning on SQA3D (Ma et al., 2023). Our evaluation metrics include conventional scores (*e.g.*, CIDEr, BLEU, METEOR, ROUGE) and other metrics adapted for open-ended generation, *e.g.*, sentence similarity (Reimers & Gurevych, 2019) and refined exact-match accuracy (see details in Appendix H.1). Following 3D-VisTA (Zhu et al., 2023c), **we use object proposals from Mask3D (Schult et al., 2022) instead of ground-truth object segments for evaluation.**

Baselines. For quantitative comparisons, we include both task-specific approaches and generalist models: 1) state-of-the-art specialists in 3D dense captioning (Chen et al., 2021; Cai et al., 2022; Chen et al., 2023); 2) state-of-the-art specialists in 3D QA (Azuma et al., 2022; Ma et al., 2023); 3) task-specific fine-tuned generalist models like 3D-VisTA (Zhu et al., 2023c) and 3D-LLM (Hong et al., 2023). To the best of our knowledge, LEO *is the first model that, in stark contrast to prior models, can directly handle the aforementioned 3D VL tasks in a unified architecture without task-specific fine-tuning*. This lends greater credence to LEO's comparative superiority.

Results & analysis. As shown in Tab. 4, LEO surpasses both state-of-the-art single-task and task-specific fine-tuned models significantly on 3D dense captioning and 3D QA tasks. In contrast to the specialist models that utilize task-specific heads, our LLM-based approach not only affords the flexibility of generating open-ended responses but also exhibits excellent quantitative results. On the other hand, considering the complicated feature aggregation in 3D-LLM, we believe that object-centric 3D representation is a simple yet effective option to connect 3D scenes with LLM while harnessing the inherent knowledge of LLM.

4.2. Scene-grounded Dialogue and Planning

Overview. Upon the 3D VL understanding and reasoning, we anticipate LEO to support more sophisticated interaction with humans, *e.g.*, responding to complex multi-round user instructions in the 3D world. To verify these capabilities, we conduct qualitative studies on 3D dialogue and planning tasks, with unseen scenarios from the held-out test sets of LEO-instruct. We defer the quantitative results of dialogue and planning to our ablation study in Sec. 4.4. Quantitative comparison with other approaches is infeasible given the absence of comparable benchmarks.

Results & analysis. As shown in Fig. A.1, LEO is capable of generating high-quality responses, which encompass two features: **1) Precisely grounded to the 3D scenes.** The task plan proposed by LEO involves concrete objects related to the 3D scene, as well as plausible actions regarding these objects. **2) Rich informative spatial relations.** The entities in LEO's responses often accompany detailed depictions. Such information helps identify specific objects in complex 3D scenes and affords considerable assistance to humans.

4.3. Embodied Action in 3D World

Overview. To probe LEO's capacity of bridging vision-language-acting in the 3D world, we select two canonical embodied AI tasks: embodied navigation (`ObjNav`) on AI Habitat (Ramrakhya et al., 2022) and robotic manipulation on CLIPort (Shridhar et al., 2021). Specifically, for CLIPort robotic manipulation, we evaluate LEO on the three tasks listed in Tab. 5 including their unseen counterparts, and report the success scores. For `ObjNav`, we evaluate LEO on the original MP3D `ObjNav` validation split. Ad-

Table 7: Quantitative results of LEO trained with different data configurations. *w/o Align*: without alignment stage. *ScanNet*: tuned on ScanNet scenes only. *w/o Act*: tuned without embodied acting tasks. We report the exact match metrics for QA tasks and sentence similarity for others. Underlined figures indicate zero-shot results on novel scenes (3RScan).

	ScanNet			3RScan		
	Scan2Cap	ScanQA	SQA3D	3RQA	3RDialog	3RPlan
w/o Align	62.8	22.7 (45.0)	**50.9** (**53.2**)	49.7 (53.7)	73.0	80.3
ScanNet	64.0	24.4 (**49.2**)	46.8 (49.5)	35.8 (50.0)	25.5	23.4
w/o Act	**65.4**	24.3 (48.5)	50.0 (52.5)	**51.9** (**57.4**)	**73.3**	**81.1**
VLA	65.3	**25.0** (48.9)	46.2 (48.3)	51.3 (55.8)	72.3	77.2

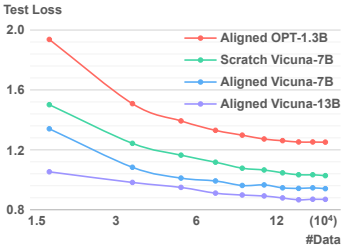
Table 8: TrueSkill scores with human preference. *Dialg*: dialogue and planning data.

	Answerable	Unanswerable	NLP
w/o Dialg	24.4±1.3	23.1±1.4	23.4±1.4
w/ Dialg	**25.6±1.3**	**26.8±1.4**	**26.6±1.4**

Table 9: Answer accuracy (EM) on object-existence questions. *Aug*: augmented data.

	3RScan			ScanNet (0-shot)		
	Yes	No	Overall	Yes	No	Overall
w/o Aug	**1.00**	0.01	0.34	**0.98**	0.16	0.43
w/ Aug	0.72	**0.91**	**0.85**	0.88	**0.81**	**0.83**

Figure 3: LEO-instruct test loss with the growth of data and model scale, manifesting the scaling law.



ditionally, we test generalization to the validation split of the newly introduced HM3D `ObjNav` task (Ramakrishnan et al., 2021). We report the success rate and SPL metrics following Ramrakhya et al. (2022). We consider both Habitat-web (Ramrakhya et al., 2022) (fully supervised) and ZSON (Majumdar et al., 2022) (zero-shot) as baselines.

Results & analysis. We present the results of CLIPort manipulation and object navigation in Tabs. 5 and 6. Our findings are as follows: 1) In robotic manipulation, LEO is comparable to state-of-the-art performances and even better on some challenging unseen tasks. In particular, LEO directly produces motor commands without inductive bias (*e.g.*, heatmap) that benefit previous methods, showcasing LEO's considerable capacity for learning embodied actions. 2) In `ObjNav`, LEO achieves a success rate that is comparable to the baselines and has a better SPL on MP3D-val, suggesting that LEO can leverage the object-centric 3D scene input (potentially offering a coarse global map) and take a shorter path to the target. Furthermore, results on HM3D-val confirm LEO's zero-shot generalization to novel scenes. Notably, all baselines are equipped with recurrent modules while LEO only incorporates truncated past actions, which could account for a lower success rate (see discussion in Appendix H.2). 3) Overall, the two-stage learning scheme endows LEO with semantic-level generalization (novel objects, *etc.*) in both manipulation and navigation tasks. We demonstrate the efficacy of tackling embodied acting tasks with a general framework from 3D VL.

Additional results. We further investigate the perception modules, data regime, and generalization to unseen objects in `ObjNav` task. See the results in Appendix I.4.

4.4. More Insights into LEO

Overview. In this section, we aim to offer deeper insights into LEO's characteristics, mainly from the data perspective (model perspective is deferred to Appendix G.2). Specifically, we evaluate LEO when trained with different data configurations, including exact match, sentence similarity, and human rating. We regard LEO instruction-tuned without embodied acting tasks (*w/o Act*) as the default setting. Following Achlioptas et al. (2020), we use ground-truth object segments in these analyses. We present additional analyses on data in Appendix I.2 and model in Appendix I.3.

Alignment stage. In contrast to complete two-stage training (*w/o Act*), we direct instruction-tune a model without alignment stage (*w/o Align*). The results in Tab. 7 show the consistent impact of alignment. In particular, the benefit of alignment is significant on Scan2Cap since it concerns detailed scene understanding and captioning, which is a primary focus of alignment training.

Specialist *vs*. generalist. We train a specialist on ScanNet scenes (*ScanNet*). As shown in Tab. 7, *ScanNet* performs slightly worse than *w/o Act* even on ScanNet tasks, and particularly struggles at generalization across scenes (3RQA) and tasks (3RDialog and 3RPlan). This demonstrates the advantage of generalist-style instruction tuning with broad coverage of scenes and tasks.

VL *vs*. VLA. We compare *w/o Act* and *VLA*, which differ in whether embodied acting tasks are included for training. The results in Tab. 7 show that incorporating embodied acting tasks could lead to performance drops on 3D VL tasks. This may stem from 1) the gap between language generation and embodied action prediction, and 2) the imbalanced data scale of embodied acting tasks. In contrast to the finding that VL data benefits embodied acting tasks in VLA co-training (Brohan et al., 2023), our observation implies that embodied acting tasks may harm VL capabilities in turn. How to continually bridge the gap between VL and embodied acting tasks is an important direction for further exploration.

Dialogue and planning data. In contrast to the default model (*w/ Dialg* in Tab. 8), we train LEO without dialogue and planning data (*w/o Dialg*). We design an evaluation set with three types of questions (Answerable, Unanswerable, and NLP) and evaluate with TrueSkill (Graepel et al., 2007) according to human preference (see details in Appendix G.3). The results in Tab. 8 confirm more hallucinations (less preferred by users on "Unanswerable") and worse NLP skills for *w/o Dialg*. This is probably because 1) the diverse conversations in our dialogue data can help cultivate flexible responses to complex instructions, and 2)

our planning data can offer scene-grounded commonsense knowledge and also encourage detailed coherent text.

Data balancing. We find imbalanced data could induce hallucination in LEO, *e.g.*, it tends to respond with "Yes" when asked "Is there something in this room?". To address this, we augment the 3RScanQA data with more negative samples where non-existent objects are queried. We also design an evaluation set with different types (Yes and No) of object-existence questions (see details in Appendix G.4). Results in Tab. 9 demonstrate that we can effectively mitigate the hallucination problem by balancing the tuning data. Moreover, the benefit of augmenting 3RScan data can transfer to ScanNet scenes in a zero-shot manner.

4.5. Scaling Law Analysis

Settings. We study the scaling effect (Kaplan et al., 2020; Reed et al., 2022) of data and model in LEO by tracking the instruction-tuning loss on the test set with the growth of data scale. In addition to the default Vicuna-7B, we incorporate two LLMs at different scales: OPT-1.3B (Zhang et al., 2022) and Vicuna-13B (Chiang et al., 2023). For Vicuna-7B, we also probe the influence of alignment (Scratch *vs*. Aligned).

Results & analysis. From the test loss curves in Fig. 3, we have the following findings: **1) The instruction tuning of LEO conforms to the scaling law** (Kaplan et al., 2020; Reed et al., 2022). We observe that all curves decrease log-linearly with the data scale. **2) Scaling up LLM leads to consistent improvements.** Aligned Vicuna-7B shows significantly lower losses than Aligned OPT-1.3B. In contrast, despite the consistent improvements, the gap between Aligned Vicuna-7B and Vicuna-13B appears less significant, suggesting potential saturation if we continue to scale up the LLM. This indicates the scalability of LEO and the necessity of scaling up data to match the model capacity. **3) Alignment leads to consistent improvements.** Aligned Vicuna-7B shows consistently lower losses than Scratch Vicuna-7B, which corresponds to the inferior performances of *w/o Align* in Tab. 7.

5. Related Work

Generalist agents. The AI community has witnessed the rising generalist models in both vision (Lu et al., 2023; Wang et al., 2023b; Kirillov et al., 2023) and language (OpenAI, 2022; 2023) domains. A generalist agent requires additional embodiment knowledge to interact with the environment and complete embodied acting tasks. Existing efforts towards generalist agents include: grounded reasoning and task planning in the real world (Ahn et al., 2022; Huang et al., 2022b), skill generalization in open-world environment (Fan et al., 2022; Cai et al., 2023a; Wang et al., 2023d;a; Cai et al., 2023b; Gong et al., 2023b), general robotic manipulation (Brohan et al., 2022; Jiang et al., 2023; Gong et al., 2023a), and unified vision-language-action (VLA) models such as Gato (Reed et al., 2022), PaLM-E (Driess et al., 2023), EmbodiedGPT (Mu et al., 2023), and RT-2 (Brohan et al., 2023). LEO belongs to the VLA model, however, its goal is to build a generalist agent that can understand the real 3D world beyond 2D images, which is absent in existing works.

Multi-modal instruction tuning. Pre-trained LLMs demonstrated practical for solving vision-language tasks (Tsimpoukelli et al., 2021; Alayrac et al., 2022; Guo et al., 2023; Li et al., 2023d; Zhao et al., 2023). Meanwhile, the instruction-tuning paradigm exhibited strong zero-shot generalization in NLP tasks (Wei et al., 2022; Sanh et al., 2022; Ouyang et al., 2022; Chung et al., 2022). The two streams merged into instruction-tuned LVLMs (Liu et al., 2023b; Zhu et al., 2023b; Ye et al., 2023; Gao et al., 2023; Li et al., 2023b; Gong et al., 2023c; Dai et al., 2023). Despite the burst, these models are confined to 2D visual modalities, *e.g.*, image or video. Concurrent works (Yin et al., 2023; Hong et al., 2023; Wang et al., 2023e; Xu et al., 2023) extend to 3D vision tasks, but these models either lack the acting capability or unified efficient architecture.

Grounded 3D scene understanding. One key obstacle to building LEO is grounding the 3D world with natural languages. There exist diverse methods of grounded scene understanding, *e.g.*, spatial relation modeling (Zhao et al., 2021; Chen et al., 2022; Zhu et al., 2023c) and fine-grained open-scene understanding (Peng et al., 2023b; Kerr et al., 2023). However, due to data scarcity, how to utilize LLMs to ground the 3D scene is rarely explored. Recently, 3D-LLM (Hong et al., 2023) leverages multi-view images and Chat-3D (Wang et al., 2023e) uses object-centric point clouds to enable the LLMs with 3D grounding. In this work, we devise both 2D and 3D encoders for grounding various visual representations and employ LoRA (Hu et al., 2022) to efficiently fine-tune the LLMs.

3D data prompting from LLMs. LLMs exhibit extraordinary capabilities of text generation and serve as a source for collecting diverse instruction-following data (Wang et al., 2023c; Taori et al., 2023; Peng et al., 2023a). However, the lack of access to visual modalities makes it troublesome to collect visual instruction-tuning data. To address this issue, existing methods provide bounding boxes (Liu et al., 2023b) and add dense captions (Li et al., 2023a; Liu et al., 2023a) as image descriptions or directly use off-the-shelf large vision-language models (LVLM) (Zhu et al., 2023a; Luo et al., 2023) to help collect such data. Unlike concurrent attempts (Yin et al., 2023; Hong et al., 2023; Wang et al., 2023e) in collecting 3D instruction-tuning data, our approach features a scene-graph-based prompting and refinement method to prompt and correct the data.

6. Conclusions

The proposed agent LEO extends the current generalist ability of LLMs from text towards the 3D world and embodied tasks. It is a crucial initial step towards building embodied generalist agents. Nonetheless, there are also limitations, *e.g.*, generalization to novel scenes, and a notable gap between VL learning and embodied action control. In light of this work, we identify several promising directions that hold the potential for substantial advancement: (1) enhancing the 3D VL understanding capability by leveraging larger-scale VL data from richer 3D domains; (2) continually bridging the gap between 3D VL and embodied action, as our experiments reveal the efficacy of their joint learning; (3) investigating the issues of safety and alignment in the context of embodied generalist agents, particularly given that our scaling law analysis suggests significant enhancements through scaling on data and model.

Impact Statement

This work introduces LEO, an embodied multi-modal generalist agent designed to extend machine learning capabilities into the 3D realm, marking a significant advance in the field. The potential societal implications of LEO are manifold, touching on robotics, AR/VR, assistive technologies, and environmental planning. Ethically, it underscores the importance of responsible AI development, emphasizing safety, privacy, and fairness in automated decision-making. As LEO ventures into new territories of human-machine interaction, it prompts a re-evaluation of ethical frameworks to ensure that advancement contributes positively to society. While the immediate societal consequences of our work align with the goals of advancing machine learning, we acknowledge the necessity of ongoing ethical consideration as applications of LEO evolve.

Acknowledgements

This work is supported in part by the National Science and Technology Major Project (2022ZD0114900).

References

Achlioptas, P., Abdelreheem, A., Xia, F., Elhoseiny, M., and Guibas, L. Referit3d: Neural listeners for fine-grained 3d object identification in real-world scenes. In *European Conference on Computer Vision (ECCV)*, 2020. 4, 7, 14

Ahn, M., Brohan, A., Brown, N., Chebotar, Y., Cortes, O., David, B., Finn, C., Fu, C., Gopalakrishnan, K., Hausman, K., et al. Do as i can, not as i say: Grounding language in robotic affordances. *arXiv preprint arXiv:2204.01691*, 2022. 8

Alayrac, J.-B., Donahue, J., Luc, P., Miech, A., Barr, I., Hasson, Y., Lenc, K., Mensch, A., Millican, K., Reynolds, M., et al. Flamingo: a visual language model for few-shot learning. *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. 1, 3, 8

Antol, S., Agrawal, A., Lu, J., Mitchell, M., Batra, D., Zitnick, C. L., and Parikh, D. Vqa: Visual question answering. In *International Conference on Computer Vision (ICCV)*, 2015. 15

Azuma, D., Miyanishi, T., Kurita, S., and Kawanabe, M. Scanqa: 3d question answering for spatial scene understanding. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2, 6, 15

Bang, Y., Cahyawijaya, S., Lee, N., Dai, W., Su, D., Wilie, B., Lovenia, H., Ji, Z., Yu, T., Chung, W., et al. A multitask, multilingual, multimodal evaluation of chatgpt on reasoning, hallucination, and interactivity. *arXiv preprint arXiv:2302.04023*, 2023. 4

Bommasani, R., Hudson, D. A., Adeli, E., Altman, R., Arora, S., von Arx, S., Bernstein, M. S., Bohg, J., Bosselut, A., Brunskill, E., et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021. 1

Brohan, A., Brown, N., Carbajal, J., Chebotar, Y., Dabis, J., Finn, C., Gopalakrishnan, K., Hausman, K., Herzog, A., Hsu, J., et al. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022. 1, 8

Brohan, A., Brown, N., Carbajal, J., Chebotar, Y., Chen, X., Choromanski, K., Ding, T., Driess, D., Dubey, A., Finn, C., et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*, 2023. 1, 3, 7, 8, 30

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 1, 3

Cai, D., Zhao, L., Zhang, J., Sheng, L., and Xu, D. 3djcg: A unified framework for joint dense captioning and visual grounding on 3d point clouds. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 6

Cai, S., Wang, Z., Ma, X., Liu, A., and Liang, Y. Open-world multi-task control through goal-aware representation learning and adaptive horizon prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13734–13744, 2023a. 8

Cai, S., Zhang, B., Wang, Z., Ma, X., Liu, A., and Liang, Y. Groot: Learning to follow instructions by watching gameplay videos. *arXiv preprint arXiv:2310.08235*, 2023b. 8

Chen, D. Z., Chang, A. X., and Nießner, M. Scanrefer: 3d object localization in rgb-d scans using natural language. In *European Conference on Computer Vision (ECCV)*, 2020. 1

Chen, S., Guhur, P.-L., Tapaswi, M., Schmid, C., and Laptev, I. Language conditioned spatial relation reasoning for 3d object grounding. *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. 1, 3, 8, 25, 26

Chen, S., Zhu, H., Chen, X., Lei, Y., Yu, G., and Chen, T. End-to-end 3d dense captioning with vote2cap-detr. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 6

Chen, S., Chen, X., Zhang, C., Li, M., Yu, G., Fei, H., Zhu, H., Fan, J., and Chen, T. Ll3da: Visual interactive instruction tuning for omni-3d understanding, reasoning, and planning. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 31

Chen, Z., Gholami, A., Nießner, M., and Chang, A. X. Scan2cap: Context-aware dense captioning in rgb-d scans. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2, 4, 6, 15

Chiang, W.-L., Li, Z., Lin, Z., Sheng, Y., Wu, Z., Zhang, H., Zheng, L., Zhuang, S., Zhuang, Y., Gonzalez, J. E., Stoica, I., and Xing, E. P. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, 2023. URL https://lmsys.org/blog/2023-03-30-vicuna/. 3, 8, 29

Chung, H. W., Hou, L., Longpre, S., Zoph, B., Tay, Y., Fedus, W., Li, E., Wang, X., Dehghani, M., Brahma, S., et al. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*, 2022. 8

Dai, A., Chang, A. X., Savva, M., Halber, M., Funkhouser, T., and Nießner, M. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 1, 15, 25

Dai, W., Li, J., Li, D., Tiong, A. M. H., Zhao, J., Wang, W., Li, B., Fung, P., and Hoi, S. Instructblip: Towards general-purpose vision-language models with instruction tuning. *arXiv preprint arXiv:2305.06500*, 2023. 8

Deitke, M., Schwenk, D., Salvador, J., Weihs, L., Michel, O., VanderBilt, E., Schmidt, L., Ehsani, K., Kembhavi, A., and Farhadi, A. Objaverse: A universe of annotated 3d objects. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 14

Driess, D., Xia, F., Sajjadi, M. S., Lynch, C., Chowdhery, A., Ichter, B., Wahid, A., Tompson, J., Vuong, Q., Yu, T., et al. Palm-e: An embodied multimodal language model. In *International Conference on Machine Learning (ICML)*, 2023. 1, 8

Fan, L., Wang, G., Jiang, Y., Mandlekar, A., Yang, Y., Zhu, H., Tang, A., Huang, D.-A., Zhu, Y., and Anandkumar, A. Minedojo: Building open-ended embodied agents with internet-scale knowledge. *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. 8

Gao, P., Han, J., Zhang, R., Lin, Z., Geng, S., Zhou, A., Zhang, W., Lu, P., He, C., Yue, X., et al. Llama-adapter v2: Parameter-efficient visual instruction model. *arXiv preprint arXiv:2304.15010*, 2023. 8

Gong, R., Huang, J., Zhao, Y., Geng, H., Gao, X., Wu, Q., Ai, W., Zhou, Z., Terzopoulos, D., Zhu, S.-C., et al. Arnold: A benchmark for language-grounded task learning with continuous states in realistic 3d scenes. In *International Conference on Computer Vision (ICCV)*, 2023a. 8

Gong, R., Huang, Q., Ma, X., Vo, H., Durante, Z., Noda, Y., Zheng, Z., Zhu, S.-C., Terzopoulos, D., Fei-Fei, L., et al. Mindagent: Emergent gaming interaction. *arXiv preprint arXiv:2309.09971*, 2023b. 8

Gong, T., Lyu, C., Zhang, S., Wang, Y., Zheng, M., Zhao, Q., Liu, K., Zhang, W., Luo, P., and Chen, K. Multimodal-gpt: A vision and language model for dialogue with humans. *arXiv preprint arXiv:2305.04790*, 2023c. 8

Graepel, T., Minka, T., and Herbrich, R. T. A bayesian skill rating system. *Advances in Neural Information Processing Systems*, 19:569–576, 2007. 7, 29

Guo, J., Li, J., Li, D., Tiong, A. M. H., Li, B., Tao, D., and Hoi, S. C. From images to textual prompts: Zero-shot vqa with frozen large language models. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 8

Hong, Y., Zhen, H., Chen, P., Zheng, S., Du, Y., Chen, Z., and Gan, C. 3d-llm: Injecting the 3d world into large language models. *arXiv preprint arXiv:2307.12981*, 2023. 1, 4, 6, 8, 20, 31

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations (ICLR)*, 2022. 3, 8, 27

Huang, J., Zhu, W. Y., Jia, B., Wang, Z., Ma, X., Li, Q., and Huang, S. Perceive, ground, reason, and act: A benchmark for general-purpose visual representation. *arXiv preprint arXiv:2211.15402*, 2022a. 1

Huang, W., Xia, F., Xiao, T., Chan, H., Liang, J., Florence, P., Zeng, A., Tompson, J., Mordatch, I., Chebotar, Y., et al. Inner monologue: Embodied reasoning through planning with language models. In *Conference on Robot Learning (CoRL)*, 2022b. 8

Jiang, Y., Gupta, A., Zhang, Z., Wang, G., Dou, Y., Chen, Y., Fei-Fei, L., Anandkumar, A., Zhu, Y., and Fan, L. Vima: General robot manipulation with multimodal prompts. In *International Conference on Machine Learning (ICML)*, 2023. 8

Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., Gray, S., Radford, A., Wu, J., and Amodei, D. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020. 2, 8

Kerr, J., Kim, C. M., Goldberg, K., Kanazawa, A., and Tancik, M. Lerf: Language embedded radiance fields. In *International Conference on Computer Vision (ICCV)*, 2023. 8

Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A. C., Lo, W.-Y., et al. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023. 1, 8

Kudo, T. and Richardson, J. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. *arXiv preprint arXiv:1808.06226*, 2018. 3

Lake, B. M., Salakhutdinov, R., and Tenenbaum, J. B. Human-level concept learning through probabilistic program induction. *Science*, 2015. 1

Lake, B. M., Ullman, T. D., Tenenbaum, J. B., and Gershman, S. J. Building machines that learn and think like people. *Behavioral and Brain Sciences*, 2017. 1

Li, B., Zhang, Y., Chen, L., Wang, J., Pu, F., Yang, J., Li, C., and Liu, Z. Mimic-it: Multi-modal in-context instruction tuning. *arXiv preprint arXiv:2306.05425*, 2023a. 8

Li, B., Zhang, Y., Chen, L., Wang, J., Yang, J., and Liu, Z. Otter: A multi-modal model with in-context instruction tuning. *arXiv preprint arXiv:2305.03726*, 2023b. 8

Li, C., Gan, Z., Yang, Z., Yang, J., Li, L., Wang, L., and Gao, J. Multimodal foundation models: From specialists to general-purpose assistants, 2023c. 1

Li, J., Li, D., Savarese, S., and Hoi, S. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597*, 2023d. 4, 8

Liu, F., Lin, K., Li, L., Wang, J., Yacoob, Y., and Wang, L. Aligning large multi-modal model with robust instruction tuning. *arXiv preprint arXiv:2306.14565*, 2023a. 8

Liu, H., Li, C., Wu, Q., and Lee, Y. J. Visual instruction tuning. *arXiv preprint arXiv:2304.08485*, 2023b. 2, 3, 4, 8

Liu, Z., Mao, H., Wu, C.-Y., Feichtenhofer, C., Darrell, T., and Xie, S. A convnet for the 2020s. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 3, 26

Lu, J., Clark, C., Zellers, R., Mottaghi, R., and Kembhavi, A. Unified-io: A unified model for vision, language, and multi-modal tasks. In *International Conference on Learning Representations (ICLR)*, 2023. 8

Luo, T., Rockwell, C., Lee, H., and Johnson, J. Scalable 3d captioning with pretrained models. *arXiv preprint arXiv:2306.07279*, 2023. 4, 8, 14

Ma, X., Yong, S., Zheng, Z., Li, Q., Liang, Y., Zhu, S.-C., and Huang, S. Sqa3d: Situated question answering in 3d scenes. In *International Conference on Learning Representations (ICLR)*, 2023. 2, 4, 6, 15, 23

Majumdar, A., Aggarwal, G., Devnani, B., Hoffman, J., and Batra, D. Zson: Zero-shot object-goal navigation using multimodal goal embeddings. *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. 7

Mountcastle, V. B. An organizing principle for cerebral function: the unit module and the distributed system. *The neurosciences. Fourth study program*, 1979. 1

Mu, Y., Zhang, Q., Hu, M., Wang, W., Ding, M., Jin, J., Wang, B., Dai, J., Qiao, Y., and Luo, P. Embodiedgpt: Vision-language pre-training via embodied chain of thought. *arXiv preprint arXiv:2305.15021*, 2023. 8

OpenAI. Chatgpt. https://openai.com/blog/chatgpt/, 2022. 1, 2, 8, 16

OpenAI. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 1, 2, 8, 20

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. 8

Peng, B., Li, C., He, P., Galley, M., and Gao, J. Instruction tuning with gpt-4. *arXiv preprint arXiv:2304.03277*, 2023a. 8

Peng, S., Genova, K., Jiang, C., Tagliasacchi, A., Pollefeys, M., Funkhouser, T., et al. Openscene: 3d scene understanding with open vocabularies. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023b. 8

Qi, C. R., Yi, L., Su, H., and Guibas, L. J. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in Neural Information Processing Systems (NeurIPS)*, 2017. 3, 25, 29

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning (ICML)*, 2021. 1, 27

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research (JMLR)*, 2020. 3

Ramakrishnan, S. K., Gokaslan, A., Wijmans, E., Maksymets, O., Clegg, A., Turner, J., Undersander, E., Galuba, W., Westbury, A., Chang, A. X., et al. Habitat-matterport 3d dataset (hm3d): 1000 large-scale 3d environments for embodied ai. *arXiv preprint arXiv:2109.08238*, 2021. 7, 30

Ramrakhya, R., Undersander, E., Batra, D., and Das, A. Habitat-web: Learning embodied object-search strategies from human demonstrations at scale. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2, 6, 7, 15, 27, 30, 32

Reed, S., Zolna, K., Parisotto, E., Colmenarejo, S. G., Novikov, A., Barth-Maron, G., Gimenez, M., Sulsky, Y., Kay, J., Springenberg, J. T., et al. A generalist agent. *Transactions on Machine Learning Research (TMLR)*, 2022. 1, 2, 8

Reimers, N. and Gurevych, I. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Annual Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2019. 6

Sanh, V., Webson, A., Raffel, C., Bach, S., Sutawika, L., Alyafeai, Z., Chaffin, A., Stiegler, A., Raja, A., Dey, M., et al. Multitask prompted training enables zero-shot task generalization. In *International Conference on Learning Representations (ICLR)*, 2022. 8

Savva, M., Kadian, A., Maksymets, O., Zhao, Y., Wijmans, E., Jain, B., Straub, J., Liu, J., Koltun, V., Malik, J., et al. Habitat: A platform for embodied ai research. In *International Conference on Computer Vision (ICCV)*, 2019. 4, 15, 30

Schmidhuber, J. One big net for everything. *arXiv preprint arXiv:1802.08864*, 2018. 1

Schuhmann, C., Beaumont, R., Vencu, R., Gordon, C., Wightman, R., Cherti, M., Coombes, T., Katta, A., Mullis, C., Wortsman, M., et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. 1, 26

Schult, J., Engelmann, F., Hermans, A., Litany, O., Tang, S., and Leibe, B. Mask3d for 3d semantic instance segmentation. *arXiv preprint arXiv:2210.03105*, 2022. 3, 6, 28, 31

Shridhar, M., Manuelli, L., and Fox, D. Cliport: What and where pathways for robotic manipulation. In *Conference on Robot Learning (CoRL)*, 2021. 2, 4, 6, 15, 27

Suglia, A., Gao, Q., Thomason, J., Thattai, G., and Sukhatme, G. Embodied bert: A transformer model for embodied, language-guided visual task completion. *arXiv preprint arXiv:2108.04927*, 2021. 30

Taori, R., Gulrajani, I., Zhang, T., Dubois, Y., Li, X., Guestrin, C., Liang, P., and Hashimoto, T. B. Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca, 2023. 8

Tsimpoukelli, M., Menick, J. L., Cabi, S., Eslami, S., Vinyals, O., and Hill, F. Multimodal few-shot learning with frozen language models. *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. 1, 8

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. *Advances in Neural Information Processing Systems (NeurIPS)*, 2017. 1, 25

Wald, J., Avetisyan, A., Navab, N., Tombari, F., and Nießner, M. Rio: 3d object instance re-localization in changing indoor environments. In *International Conference on Computer Vision (ICCV)*, 2019. 1, 15

Wang, G., Xie, Y., Jiang, Y., Mandlekar, A., Xiao, C., Zhu, Y., Fan, L., and Anandkumar, A. Voyager: An open-ended embodied agent with large language models. *arXiv preprint arXiv:2305.16291*, 2023a. 8

Wang, X., Wang, W., Cao, Y., Shen, C., and Huang, T. Images speak in images: A generalist painter for in-context visual learning. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023b. 8

Wang, Y., Kordi, Y., Mishra, S., Liu, A., Smith, N. A., Khashabi, D., and Hajishirzi, H. Self-instruct: Aligning language model with self generated instructions. In

Annual Meeting of the Association for Computational Linguistics (ACL), 2023c. 8

Wang, Z., Cai, S., Liu, A., Ma, X., and Liang, Y. Describe, explain, plan and select: Interactive planning with large language models enables open-world multi-task agents. *arXiv preprint arXiv:2302.01560*, 2023d. 8

Wang, Z., Huang, H., Zhao, Y., Zhang, Z., and Zhao, Z. Chat-3d: Data-efficiently tuning large language model for universal dialogue of 3d scenes. *arXiv preprint arXiv:2308.08769*, 2023e. 4, 8

Wei, J., Bosma, M., Zhao, V., Guu, K., Yu, A. W., Lester, B., Du, N., Dai, A. M., and Le, Q. V. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations (ICLR)*, 2022. 8

Wu, S.-C., Wald, J., Tateno, K., Navab, N., and Tombari, F. Scenegraphfusion: Incremental 3d scene graph prediction from rgb-d sequences. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 4, 17

Xu, R., Wang, X., Wang, T., Chen, Y., Pang, J., and Lin, D. Pointllm: Empowering large language models to understand point clouds. *arXiv preprint arXiv:2308.16911*, 2023. 8, 29

Ye, Q., Xu, H., Xu, G., Ye, J., Yan, M., Zhou, Y., Wang, J., Hu, A., Shi, P., Shi, Y., et al. mplug-owl: Modularization empowers large language models with multimodality. *arXiv preprint arXiv:2304.14178*, 2023. 8

Yin, Z., Wang, J., Cao, J., Shi, Z., Liu, D., Li, M., Sheng, L., Bai, L., Huang, X., Wang, Z., et al. Lamm: Language-assisted multi-modal instruction-tuning dataset, framework, and benchmark. *arXiv preprint arXiv:2306.06687*, 2023. 4, 8

Yu, X., Tang, L., Rao, Y., Huang, T., Zhou, J., and Lu, J. Point-bert: Pre-training 3d point cloud transformers with masked point modeling. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 29

Zhang, S., Roller, S., Goyal, N., Artetxe, M., Chen, M., Chen, S., Dewan, C., Diab, M., Li, X., Lin, X. V., et al. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*, 2022. 8, 29

Zhao, H., Cai, Z., Si, S., Ma, X., An, K., Chen, L., Liu, Z., Wang, S., Han, W., and Chang, B. Mmicl: Empowering vision-language model with multi-modal in-context learning. *arXiv preprint arXiv:2309.07915*, 2023. 8

Zhao, L., Cai, D., Sheng, L., and Xu, D. 3dvg-transformer: Relation modeling for visual grounding on point clouds. In *International Conference on Computer Vision (ICCV)*, 2021. 1, 8

Zhu, D., Chen, J., Haydarov, K., Shen, X., Zhang, W., and Elhoseiny, M. Chatgpt asks, blip-2 answers: Automatic questioning towards enriched visual descriptions. *arXiv preprint arXiv:2303.06594*, 2023a. 8

Zhu, D., Chen, J., Shen, X., Li, X., and Elhoseiny, M. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*, 2023b. 8

Zhu, Y., Gao, T., Fan, L., Huang, S., Edmonds, M., Liu, H., Gao, F., Zhang, C., Qi, S., Wu, Y. N., et al. Dark, beyond deep: A paradigm shift to cognitive ai with humanlike common sense. *Engineering*, 2020. 1

Zhu, Z., Ma, X., Chen, Y., Deng, Z., Huang, S., and Li, Q. 3d-vista: Pre-trained transformer for 3d vision and text alignment. In *International Conference on Computer Vision (ICCV)*, 2023c. 1, 3, 4, 6, 8, 14, 22

A. Qualitative Results

Figure A.1: **Qualitative results of interacting with LEO** on unseen scenarios from a held-out test set of LEO-instruct. LEO's responses and actions can be grounded in novel scenes.

B. Data

B.1. More Details on LEO-align

Object-level caption. To facilitate object-level grounding of detailed object attributes, we leverage Cap3D (Luo et al., 2023), which contains language descriptions for objects in Objaverse (Deitke et al., 2023). Given a single 3D object as input, LEO will be asked to predict its caption.

Object-in-the-scene caption. For a better understanding of how an object can be related to others (spatial relations, *etc*.) when situated in a 3D scene, we collect referring expressions of objects in scenes from existing datasets, including ScanScribe (Zhu et al., 2023c) and ReferIt3D (Achlioptas et al., 2020). Further, we generate additional object-referring

expressions on 3RScan (Wald et al., 2019) scenes by prompting LLMs (details in Appendix B.4). During alignment, LEO needs to predict these referring expressions given the object-centric 3D input of the scene and the referred object.

Scene-level caption. Finally, we encourage LEO to capture scene-level descriptions of a 3D scene. These scene-level captions focus on global information depicting key objects in the scene as well as their attributes and functionalities, relations among multiple objects, and room types and styles. We leverage scene graph annotations (Wald et al., 2019) and prompt LLMs to produce a total of ~20K captions. To further increase caption diversity, we propose a subgraph sampling strategy to prevent LLMs from always attending to certain notable facets of the scene (details in Appendix B.7). Similar to previous settings, LEO needs to predict these captions given the corresponding 3D input.

B.2. More Details on LEO-instruct

Below, we provide a comprehensive illustration of the data preparation process for these tasks and an overview of generated data in Fig. 2. We list the corresponding instructions in Appendix C.

3D captioning. The task is to produce a generic caption given 3D input. We adopt the Scan2Cap dataset (Chen et al., 2021), which is based on the ScanNet (Dai et al., 2017) 3D scenes and covers various levels (object-level and scene-level) and aspects (attributes, relations, *etc*.) of scene details.

3D question answering. The 3D-QA task is an extension of VQA (Antol et al., 2015) to 3D scenes with a focus on 3D knowledge, ranging from spatial relations to functionalities of objects. For this task, we first aggregate two existing 3D-QA datasets: ScanQA (Azuma et al., 2022) and SQA3D (Ma et al., 2023). To further generate questions concerning rich 3D knowledge, we prompt LLMs to generate ~35K QA pairs on 3RScanQA with our quality refinement techniques discussed in Sec. 3.3.

3D dialogue. The goal of this task is to support natural conversations between LEO and users about a given 3D scene. This task necessitates coherence and continuity across multiple rounds of conversational interactions. We build such dialogues on 3RScan scenes by prompting LLMs with a variant of the Chain-of-Thought prompting method discussed in Sec. 3.3 to facilitate diverse dialogues about relevant and accurate details about the 3D scene. In total, ~11K dialogues are collected.

Scene-aware task planning. In this task, LEO is required to decompose high-level tasks into step-by-step low-level plans given 3D scenes. We expect LEO to generate feasible plans based on the current 3D scene and ground its inherent common sense knowledge about procedures to the scene configurations, including, objects, their attributes, relations, and functional characteristics, *etc*. By prompting LLMs, we end up collecting ~14K task-plan pairs on 3RScan scenes.

Embodied navigation. We follow imitation learning setting in Habitat-web (Ramrakhya et al., 2022) for the embodied navigation task. We choose `ObjNav`, where LEO needs to map navigation instructions (*e.g*. "find bed"), object-centric 3D input, and an egocentric 2D input into discrete habitat motor commands. For simplicity, we use shortest path navigation trials rather than human demonstrations for learning as they are less noisy and therefore easier to learn when provided with the 3D scene. In total, we generate ~60K navigation episodes out of the MP3D `ObjNav` training scenes (Savva et al., 2019) for this task.

Robotic manipulation. We employ a subset of the manipulation tasks introduced in CLIPort (Shridhar et al., 2021). The input of this task includes instructions, egocentric 2D observations, and object-centric 3D information. As discussed in Sec. 2.1, we discretize the continuous action space of CLIPort into bins to unify the action decoding of navigation and manipulation (more details in Appendix D.3). We generate 100K demonstrations for each selected manipulation task.

B.3. Design of Seed Tasks for LLM-assisted 3D Data Generation

Object Scene Caption & Scene Caption. To align the 3D scene/object with language, we prompt ChatGPT to curate these two types of caption data. Object Scene Caption includes the spatial relationships of the object with some adjacent objects in the scene. Scene Caption is the comprehensive description for the whole 3D scene, including some key objects and their spatial relationships.

QA & Dialogue. For QA, we design several question-answer pairs given a scene graph. A diverse set of questions are asked about the 3D scene, including the object attributes, object counting, object existence, spatial relationships between the objects, object types, object affordance, room type and so on. For dialogue, we design a conversation between the assistant and a person asking questions about this scene. The answers are in a tone as if the assistant is understanding the scene and helping the person. Different from single-round QA, dialogue has some high-level tasks such as 'searching for specific

```
messages = [{''role'': ''system'', ''content'': ''You are an AI visual assistant in a 3D scene. The scene contains some objects, which compose a scene graph in json format. Each entity in the scene graph denotes an object instance, with a class label and an object id. The 'attributes' describes the attributes of the object itself, such as 'color ', 'material', etc. The 'relations' describes the spatial relations with other objects.
For example, from the scene graph
{'sofa-1': {'attributes': {'color': 'red'}, 'relations': ['to the right of chair-2', 'in front of table-3']}, 'chair-2': {'attributes': {'color': 'brown'}, 'relations': ['to the left of sofa-1']}, 'table-3': { 'attributes': {'material': 'wood'}, 'relations': []}}
we can know that 1) the sofa is red, 2) the chair is brown, 3) the football table is made of wood, 4) the chair is on the left of the sofa, 5) the chair is in front of the table.
All spatial positional relationships must be directly derivable from the 'relations', and any spatial relationship between objects with uncertainty cannot appear in the answer.

You need to generate meaningful conversations based on the scene information. The conversations include questions from human and responses from an AI assistant. Ask questions about the object types, counting the objects, object attributes, relative positions between objects. Also ask questions concerning commonsense, e.g., how the objects can be used by human and human activity in the scene. You can ask questions about the affordance of the objects in the scene. The questions should conform to the given scene information. The attributes of objects and spatial relations between objects can only be inferred from the 'attributes' and 'relations' in scene graph, respectively. The questions should contain interrogative sentences and declarative sentences to cover diverse tones. You need to first provide the context of the dialogue. The context can be high level or low level tasks. The dialogue should be related to the context. Then you need to provide the clues about the question. Then the robot answers the question according to the thought. The dialogue has the following format:Dialogue Context: <Dialogue Context>\nHuman:<Question>\nThought:<Thought>\nRobot:<Answer>. Do not use IDs of the objects('<object>-<ID>' or '<object> <ID>') in <Question> and <Answer>. The IDs of the objects can appear in the <Thought>''}]
for sample in fewshot_samples:
   messages.append({''role'': ''user'', ''content'': sample['content']})
   messages.append({''role'': ''assistant'', ''content'': sample['response']})
messages.append ({''role'': ''user'', ''content'': '\n'.join(sample['query'])})
```

Figure A.2: The prompt for generating 3D Dialogue.

Table A.1: The effect of O-CoT on the answer accuracy for Object Counting questions.

Settings	Seed 1	Seed 2	Seed 3	Seed 4	Average	Avg. Gain
w/o O-CoT	0.5838	0.5349	0.5962	0.5816	0.5741	0.2061
O-CoT	0.7647	0.8117	0.7778	0.7667	0.7802	

objects' that require multi-round conversations.

Planning. To include a deeper understanding of the global 3D scene information, we prompt ChatGPT to generate a high-level task and 5-10 action steps(interaction between the assistant and the objects in the scene) to finish the task.

B.4. Prompts for LLM-assisted 3D Data Generation

In Fig. A.2–A.6, we show the prompts for five types of LLM-assisted 3D-language data generation. We provide few-shot examples as the context. In each example, the "content" contains a scene graph, and the "response" refers to a human-labeled response. The query is a new scene graph, based on which ChatGPT (OpenAI, 2022) generates responses.

Fig. A.2 shows the prompt for generating 3D dialogue data. Red fonts outline our requirements of the dialogue content, including object attributes, spatial relations, and commonsense topics. Purple fonts formulate the template of the response. We require the response generated by the ChatGPT should include the dialogue context as well; the "thought" contains the involved objects in the question, which is used to enhance the reliability of the answer. These two components will be removed after the refinement procedures.

```
messages = [{''role'': ''system'', ''content'': ''You are an AI visual assistant in a 3D scene. The scene contains some objects, which compose a scene graph in json format. Each entity in the scene graph denotes an object instance, with a class label and an object id. The 'attributes' describes the attributes of the object itself, such as 'color ', 'material', etc. The 'relations' describes the spatial relations with other objects.
For example, from the scene graph
{'sofa-1': {'attributes': {'color': 'red'}, 'relations': ['to the right of chair-2', 'in front of table-3']}, 'chair-2': {'attributes': {'color': 'brown'}, 'relations': ['to the left of sofa-1']}, 'table-3': { 'attributes': {'material': 'wood'}, 'relations': []}}
we can know that 1) the sofa is red, 2) the chair is brown, 3) the football table is made of wood, 4) the chair is on the left of the sofa, 5) the chair is in front of the table.
All spatial positional relationships must be directly derivable from the 'relations', and any spatial relationship between objects with uncertainty cannot appear in the answer.

You need to generate 10-15 question-answer pairs based on the scene information. The question-answer pairs include the object types, counting the objects, object attributes, relative positions between objects. The questions should conform to the given scene information. The attributes of objects and spatial relations between objects can only be inferred from the 'attributes' and 'relations' in scene graph, respectively. The questions must be able to be answered correctly based on the scene graph. You need to provide the queried object. Note that all answers to the questions must be single words or phrases. The question answer pair should be following format:\nQ: <question>\nT: <queried object(s)>\nA: <Answer>. You can answer the question according to the queried object(s). If there is no information about the question, the <Answer> should be 'unkown.''}]
for sample in few_shot_samples:
    messages.append({''role'': ''user'', ''content'': sample['content']})
    messages.append({''role'': ''assistant'', ''content'': sample['response']})
messages.append ({''role'': ''user'', ''content'': '\n'.join(sample['query'])})
```

Figure A.3: The prompt for generating 3D QA.

```
messages = [{''role'': ''system'', ''content'': ''You are an AI visual assistant that can analyze a 3D scene. The scene contains some objects, which compose a scene graph in json format. Each entity in the scene graph denotes an object instance, with a class label and an object id. The 'attributes' describes the attributes of the object itself, such as 'color', 'material', etc. The 'relations' describes the spatial relations with other objects.
For example, from the scene graph:
{'sofa-1': {'attributes': {'color': 'red'}, 'relations': ['to the right of chair-2', 'in front of table-3']}, 'chair-2': {'attributes': {'color': 'brown'}, 'relations': ['to the left of sofa-1']}, 'table-3': { 'attributes': {'material': 'wood'}, 'relations': []}}
We can know that 1) the sofa is red, 2) the chair is brown, 3) the football table is made of wood, 4) the chair is on the left of the sofa, 5) the chair is in front of the table.
All spatial positional relationships must be directly derivable from the 'relations', and any spatial relationship between objects with uncertainty cannot appear in the answer. Do not use the id of the object in the dialogue, use ordinal words and attributes to refer to different objects with the same label.

Using the provided scene graph, design a high-level task that can be performed in this 3D scene. Besides, decomposing this high-level task into a sequence of action steps that can be performed using the instances in this3D scene.
Remeber, the high-level task and action steps must be able to be performed in the 3D scene using the given object instances. Do not use IDs of the objects('<object>-<ID>' or '<object> <ID>') in the planning.''}]
for sample in fewshot_samples:
    messages.append({''role'': ''user'', ''content'': sample['content']})
    messages.append({''role'': ''assistant'', ''content'': sample['response']})
messages.append ({''role'': ''user'', ''content'': '\n'.join(sample['query'])})
```

Figure A.4: The prompt for generating 3D planning.

B.5. Analysis of the Object-Centric Chain-of-Thought

To further investigate the impact of Object-centric Chain-of-Thought (O-CoT) on data quality, we analyze the answer accuracy for Object Counting questions. Specifically, we collect several demonstrations, and for each run, we select two of them as the prompt seed. With these seeds, we generate dialogues across all scenes in 3DSSG (Wu et al., 2021) and then

```
messages = [{''role'': ''system'', ''content'': ''You are an AI visual assistant in a 3D scene. The scene contains some objects, which compose a scene graph in json format. Each entity in the scene graph denotes an object instance, with a class label and an object id. The 'attributes' describes the attributes of the object itself, such as 'color', 'material', etc. The 'relations' describes the spatial relations with other objects.
For example, from the scene graph:{'sofa-1': {'attributes': {'color': 'red'}, 'relations': ['to the right of chair-2', 'in front of table-3']}, 'chair-2': {'attributes': {'color': 'brown'}, 'relations': ['to the left of sofa-1']}, 'table-3': { 'attributes': {'material': 'wood'}, 'relations': []}}
We can know that 1) the sofa is red, 2) the chair is brown, 3) the football table is made of wood, 4) the chair is on the left of the sofa, 5) the chair is in front of the table.
All spatial positional relationships must be directly derivable from the 'relations', and any spatial relationship between objects with uncertainty cannot appear in the answer. Don't use IDs of the objects('<object label>-<ID>' or '<object label> <ID>') in the summary.

You need to provide a summary for a scene. The summary should be about the object types, object attributes, relative positions between objects. Also describe the scene concerning commonsense, e.g., how the objects can be used by human and human activity in the scene. The description should conform to the given scene information. The attributes of objects and spatial relations between objects can only be inferred from the 'attributes' and 'relations' in scene graph, respectively. You don't need to describe each object in the scene, pick some objects of the scene for summary. You can also summarize the room's function, style, and comfort level based on the arrangement and color of objects within the room. Your summary must not exceed 110 words.''}]
for sample in few_shot_samples:
   messages.append({''role'': ''user'', ''content'': sample['content']})
   messages.append({''role'': ''assistant'', ''content'': sample['response']})
messages.append ({''role'': ''user'', ''content'': '\n'.join(sample['query'])})
```

Figure A.5: The prompt for generating 3D scene caption.

```
messages = [{''role'': ''system'', ''content'': ''You are a helpful assistant. You will receive a dictionary of an object. This dictionary provides information about a node in a scene graph, as well as its adjacent nodes. The value of the key 'object' is the object represented by the node. The value of 'relations' includes the spatial relationships with the adjacent nodes . The value of the key 'attribute' provides the attributes of the object. The value of 'edge attribute' provides a list of object attributes for the adjacent nodes. You need to describe the object according to the information of the target object node. The IDs of objects cannot appear in the summary.''}]
for sample in few_shot_samples:
   messages.append({''role'': ''user'', ''content'': sample['content']})
   messages.append({''role'': ''assistant'', ''content'': sample['response']})
messages.append ({''role'': ''user'', ''content'': '\n'.join(sample['query'])})
```

Figure A.6: The prompt for generating 3D object-in-the-scene caption.

assess the answer accuracy for Object Counting questions. The results are presented in Tab. A.1.

The results in Tab. A.1 indicate that O-CoT consistently improves the answer accuracy for Object Counting questions. Though there remain errors after applying O-CoT, we will conduct refinement to fix them. Examples of Object Counting questions are provided in Appendix B.6.

B.6. Refinement Details

We conduct refinement by passing raw LLM-generated responses into several human-defined filtering procedures based on the 3D scene graph. The refinement considers five raw response categories:

- Object Counting. The question concerns counting the target object.
- Object Existence. The response claims the existence of objects, which can be actually either existent or non-existent.
- Object Non-existence. The response claims the non-existence of objects, which can be actually either existent or non-existent.
- Negative Response. The scene graph cannot provide a solid response to the question, which means the question cannot be answered and will be discarded.

Table A.2: **Examples of dialogue refinement**.

Types	Raw Responses	Refined Responses
Object Counting	There are 3 chairs in the room. I see there are two washing machines in the bathroom.	There are 4 chairs in the room. I see there are 4 washing machines in the bathroom.
Object Existence	Yes, there is a cutting board in the kitchen. Yes, there is a computer and a monitor on the desk. However, the monitor is currently off.	No, there is no cutting board in the room. No, there is no computer in the room.
Object Non-existence	No, there is no stereo equipment in the room. I'm sorry, but I couldn't find a hair dryer in the bathroom.	Yes, there is a stereo equipment in the room. Yes, I found a hair dryer in the room.
Negative Response	No, there is nothing else mentioned in the scene graph. I'm sorry, but there is no mention of a mirror in the scene graph for the bathroom.	*The negative responses will be removed.*
Response with ID	You can place your backpack on the floor, to the left of the dining table-33. As for your bag, you can place it on the floor, to the left of the bed-10.	You can place your backpack on the floor, to the left of the dining table. As for your bag, you can place it on the floor, to the left of the bed.

Table A.3: **Examples of QA refinement**.

Types	Raw Responses	Refined Responses
Object Counting	Q: How many chairs are in the room? A: 3	Q: How many chairs are in the room? A: four
Object Existence	Q: Is there a mirror in the room? A: yes	Q: Is there a mirror in the room? A: no
Object Non-existence	Q: Is there an ironing board in the room? A: no	Q: Is there an ironing board in the room? A: yes
Negative Response	Q: What is the material of the bathtub? A: unknown Q: Where is the shampoo dispenser? A: unknown	*The negative responses will be removed.*
Response with ID	Q: Where is the mirror located? A: attached to wall-3, behind heater-18, to the left of shelf-19	Q: Where is the mirror located? A: attached to a wall, behind a heater, to the left of a shelf

- Response with ID. The response contains unexpected object IDs.

Specifically, we employ regular expression matching to detect errors in these five categories. We also employ this method to correct the responses except for Response with ID, which will be rewritten by ChatGPT instead. The QA pair will be eliminated if multiple rounds of rewriting fail to remove the IDs. Tab. A.2 and Tab. A.3 show some examples of the responses subject to the above five categories as well as the effect of our refinement.

B.7. Subgraph Sampling

To enhance the diversity of the 3D scene graphs used for prompting, we perform subgraph sampling on the 3DSSG according to a sampling rate, which denotes the ratio of preserved nodes. The sampled subgraphs are used for generating scene captions and planning data. We analyze the distribution of node numbers across the 3DSSG dataset in Fig. A.7 and set different sampling rates for scenes with different numbers of nodes in Tab. A.4. For each sampling rate, we set 4 random

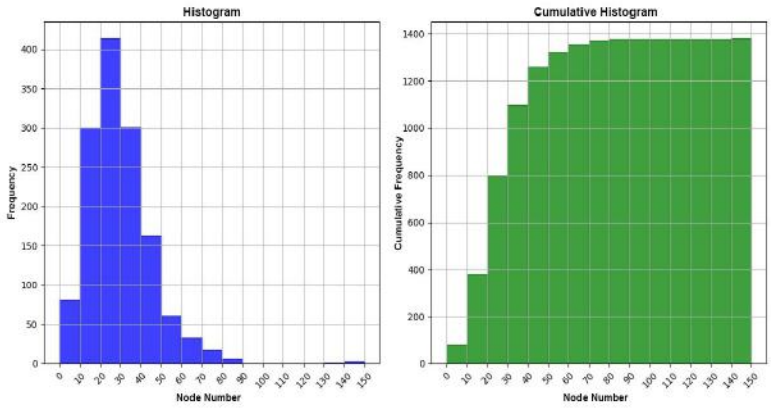


Figure A.7: **The distribution of node numbers for 3DSSG scenes.** The node number represents the number of objects in a scene.

Table A.4: **Sampling rates for scenes with different node numbers.** The hyphen denotes a sweep of sampling rates, *e.g.*, "0.7-0.9" means "0.7,0.8,0.9".

Node Number	10-20	20-30	30-40	40-50	50-60	60-70	>70
Sampling Rate	0.8,0.9	0.7-0.9	0.6-0.9	0.6-0.9	0.5-0.9	0.5-0.9	0.4-0.9

Box-based Content

wall:[-0.66, 0.853, -0.329], floor:[0.291, 0.454, -1.533], ceiling:[0.3, 0.955, 0.9], wall:[0.997, 0.577, -0.353], light:[0.213, 0.59, 0.905], wall:[0.971, 3.168, -0.351], window:[0.943, 3.385, 0.074], board:[-0.649, -0.117, -1.183], desk:[0.696, 2.259, -0.987], box:[-0.395, 0.64, -1.33], bowl:[0.631, 3.071, -0.803], box:[0.797, 3.121, -0.91]

Scene-Graph-based Content

{'wall-1': {'relations': ['attached to floor-2'], 'attribute': {'shape': 'flat', 'lexical': 'architectural', 'color': 'white'}}, 'floor-2': {'relations': [], 'attribute': {'material': 'plastic', 'shape': 'flat', 'lexical': 'inside', 'color': 'blue'}}, 'ceiling-3': {'relations': ['attached to wall-1', 'attached to wall-4', 'attached to wall-7'], 'attribute': {'shape': 'flat', 'lexical': 'overhead', 'color': 'white'}}, 'wall-4': {'relations': ['attached to floor-2'], 'attribute': {'shape': 'flat', 'lexical': 'architectural', 'color': 'white'}}, 'light-6': {'relations': ['hanging on ceiling-3'], 'attribute': {'state': 'off'}}, 'wall-7': {'relations': ['attached to floor-2'], 'attribute': {'shape': 'flat', 'lexical': 'architectural', 'color': 'white'}}, 'window-8': {'relations': ['attached to wall-7', 'behind desk-10'], 'attribute': {'material': 'glass', 'color': 'dark', 'shape': 'rectangular', 'state': 'closed'}}, 'board-9': {'relations': ['lying on floor-2', 'to the left of desk-10', 'close by box-11'], 'attribute': {'shape': 'flat', 'lexical': 'flat', 'color': 'brown'}}, 'desk-10': {'relations': ['standing on floor-2', 'in front of window-8', 'to the right of board-9', 'to the right of box-11', 'close by box-11'], 'attribute': {'other': 'rigid', 'size': 'narrow'}}, 'box-11': {'relations': ['standing on floor-2', 'close by board-9', 'close by desk-10', 'to the left of desk-10', 'in front of box-15', 'to the left of box-15'], 'attribute': {'state': 'written on', 'shape': 'rectangular', 'lexical': 'rectangular', 'other': 'rigid', 'size': 'tall'}}, 'bowl-14': {'relations': [], 'attribute': {}}, 'box-15': {'relations': ['standing on desk-10', 'to the right of box-11', 'behind box-11'], 'attribute': {'color': 'dark', 'shape': 'rectangular', 'lexical': 'rectangular', 'size': 'small'}}}

Figure A.8: Comparison of the content between box-based and scene-graph-based prompting.

prompt seeds to further enhance the diversity of prompted data.

To verify whether the subgraph sampling strategy can maintain the consistency and diversity of scene captions, we generate scene captions for the same scene using both the full graph and subgraph. We then employ GPT-4 (OpenAI, 2023) to evaluate the similarities and differences between the two captions. The results in Tab. A.5 indicate that our subgraph sampling strategy can maintain both consistency and diversity.

B.8. Scene-graph-based Prompting *vs*. Box-based Prompting

In this section, we provide a comparative analysis of scene-graph-based prompting and box-based prompting (Hong et al., 2023). We refer the readers to Figure 6 in 3D-LLM (Hong et al., 2023) for details of the box-based prompting method. Fig. A.8 shows the contents of two methods. To present a fair comparison between the two methods, we prompt with 1) demonstrations that have similar content under the same scene (see Fig. A.9) and 2) identical new scene queries. Since

Table A.5: **The comparative analysis of full graph prompting and subgraph prompting.**

Settings	Generated Scene Captions
Full Graph	In this scene, there is a colorful and flat white wall. A small TV stand is positioned in front of a low upholstered couch, to the left of a brown lamp, and to the right of a wooden table. The couch is covered with a pink padded square pillow and a patterned blanket. A light hangs on the wall and another light is supported by the table. There is a rectangular glass window attached to the wall, behind which is a yellow curtain. The room has a comfortable and cozy atmosphere, suitable for relaxation and entertainment.
Sampling Rate 0.6	In this scene, there is a colorful white wall attached to a brown wooden floor. The couch is upholstered and low in size, located to the right of a small TV stand and a table. A red blanket covers the couch, and there are patterned, padded square pillows on the couch. A black glass window is attached to the wall, and there are artificial lamps standing on the floor and close to the couch. The lamps are also near a table and to the right of the TV stand. The scene suggests a comfortable living room with a cozy couch, pillows, and a blanket, suitable for relaxation and entertainment.
GPT-4 Evalutaion	**Similarities:** 1. Both scenes describe a living room setting, emphasizing comfort and coziness suitable for relaxation and entertainment. 2. Both rooms have a white wall. 3. Each scene features a couch and a TV stand. 4. Both scenes describe a blanket and a padded square pillow on the couch. **Differences:** 1. The first scene has a brown wooden floor, while the floor is not mentioned in the second scene. 2. The first scene has a red blanket on the couch; the second has a patterned blanket but doesn't specify the color. 3. The first scene describes the lamps as "standing on the floor", while the second mentions one light hanging on the wall and another supported by the table. 4. The second scene includes a yellow curtain behind the window, which the first scene does not mention. **Summary:** Overall, both summaries provide a similar thematic view of a comfortable living room but differ in the specific arrangement and color details of the items within the room.

Box-based Prompting

Demonstration #1:
Human: Do you think the room is suitable for me to read a book?
Robot: Let me check the arrangement of the room. There is a light in the room, which can provide adequate lighting. There is a desk for you to put your book. However, there is no chair of sofa in the room. It means you cannot seat down and rest in the room.
Human: Thanks a lot. Can you describe the position of the desk?
Robot: Sure! The desk is in front of the window.
Human: Thanks a lot.
Robot: You are welcome!
Demonstration #2:
Human: Can you help me print a document?
Robot: Sure! There is a printer in the room. You can use it for printing.
Human: Ok. I also need a laptop to download the document.
Robot: Ok. I find a laptop on the table. Do you need any other help?
Human: Thanks a lot.
Robot: You are welcome!

Scene-Graph-based Prompting

Demonstration #1:
Dialogue Context: high level task: verify if the furniture arrangement is suitable for reading. low level task: check specific object.
Human: Do you think the room is suitable for me to read a book?
Thought:light-6,desk-10
Robot: Let me check the arrangement of the room. There is a light in the room, which can provide adequate lighting. There is a desk for you to put your book. However, there is no chair of sofa in the room. It means you cannot seat down and rest in the room.
Human: Thanks a lot. Can you describe the position of the desk?
Thought: desk-10
Robot: Sure! The desk is in front of the window.
Human: Thanks a lot.
Thought: no specific object
Robot: You are welcome!
Demonstration #2:
Dialogue Context: high level tasks: find proper objects to print a document.
Human: Can you help me print a document?
Thought: printer-30
Robot: Sure! There is a printer in the room. You can use it for printing.
Human: Ok. I also need a laptop to download the document.
Thought: laptop-28
Robot: Ok. I find a laptop on the table. Do you need any other help?
Human: Thanks a lot.
Thought: no specific object
Robot: You are welcome!

Figure A.9: The demonstrations for prompting, which are designed to be similar for a fair comparison.

3D-LLM does not elaborate on attribute-related prompts, we mainly compare the spatial relations in the responses. As

Box-based Prompting

Response:
Human: Can you help me find a towel?
Robot: Of course! I found three towels in the room. One is near the bathtub, one is near the sink, and another one is near the stool. Which one would you like to use?
Human: The one near the sink, please.
Robot: Alright. I will fetch the towel near the sink for you. Is there anything else I can assist you with?
Human: No, thank you. That's all.
Robot: You're welcome! If you need any further assistance, feel free to ask.

Scene-Graph-based Prompting

Response:
Dialogue Context: high level task: describe the objects in the bathroom.
Human: Can you describe the objects in the bathroom?
Thought: bathtub-8, sink-12, toilet-15, mirror-4
Robot: Sure! In the bathroom, there is a bathtub made of ceramic, with a shiny texture and a white color. There is also a sink with an oval shape and a shiny texture. The toilet is white and seat down. There is a mirror attached to the wall, which is polished. Is there anything specific you want to know about these objects?
Human: Is there a towel in the bathroom?
Thought: towel-10
Robot: Yes, there is a folded together rectangular towel in the bathroom. It is located to the right of the door.

Figure A.10: The responses of two prompting methods. Descriptions highlighted in red show our method leads to more flexible and reliable spatial relations.

shown in Fig. A.10, we highlight some spatial relations in red. The comparison shows that our method provides more diverse and reliable spatial relations, which are important for 3D scene understanding.

B.9. Additional Comparision Regarding Dataset Quality

In addition to assessing the factual accuracy of responses compared to 3D-LLM, we also compared the grammatical correctness of the responses with ScanScribe(Zhu et al., 2023c), a template-based synthetic dataset that focuses on 3D object caption. We observed that their dataset exhibited some grammar errors, whereas our dataset did not manifest such issues. We provide some data examples in Tab. A.6 and Tab. A.7. We highlighted the grammar errors present in ScanScribe dataset in red. Through comparison, it is evident that our sentences exhibit accurate and natural syntax, and also surpasses ScanScribe in the diversity and complexity of object descriptions.

Table A.6: Object captions in the 3Rscan scene 8f0f144b-55de-28ce-8053-2828b87a0cc9.

object label-id	method	response id	caption
microwave-8	ours	1	The microwave is black in color and is standing on the table. It is located behind and to the left of the toaster.
		2	The microwave is black in color. It is standing on a table and is positioned behind and to the left of a toaster.
	ScanScribe	1	This is a black microwave. It is standing on the table. It is close by the toaster. A black microwave is in front of the this microwave.
		2	This is a black microwave. It is behind the toaster. It is standing on the table. A black microwave is in front of the this microwave.

B.10. Dataset Statistics

We provide statistics on the instruction-tuning datasets. We visualize the distribution of the question types in 3RQA (Fig. A.11) and 3RDialog (Fig. A.12). The pie chart's inner circle represents the first word of the questions, while the outer circle accounts for the second or third word in the corresponding questions. The results show that the questions cover the attributes and spatial relations of the objects, as well as high-level topics such as room types and functionalities.

We also provide statistics of the root noun-verb pairs for instructions and responses in 3RDialog and 3RPlan, as shown in Fig. A.13–A.16.

C. Data Examples

Please refer to Tabs. A.25–A.27 for examples of our dataset.

Table A.7: Object captions in the 3Rscan scene 0cac753c-8d6f-2d13-8e27-e0664fc33bb9.

object label-id	method	response id	caption
counter-4	ours	1	The counter is standing on a wooden floor. It is close to 4 stools and an armchair. The stools are simple and wide in size, while the armchair is narrow in size. The floor is wooden and flat, and it is clean.
		2	The counter is standing on the wooden floor. It is close to 4 stools and an armchair. The floor is made of wooden material and is flat in shape. It is also clean. The armchair is narrow in size while the stools are simple in design. Three of the stools are wide in size.
	ScanScribe	1	This is a counter. It is close by the armchair. It is close by the stool. A counter is close by the this counter. A counter is close by the this counter.

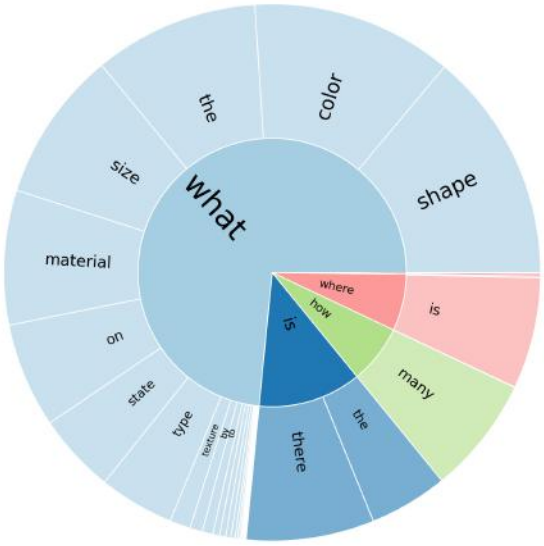


Figure A.11: Question types: 3RQA.

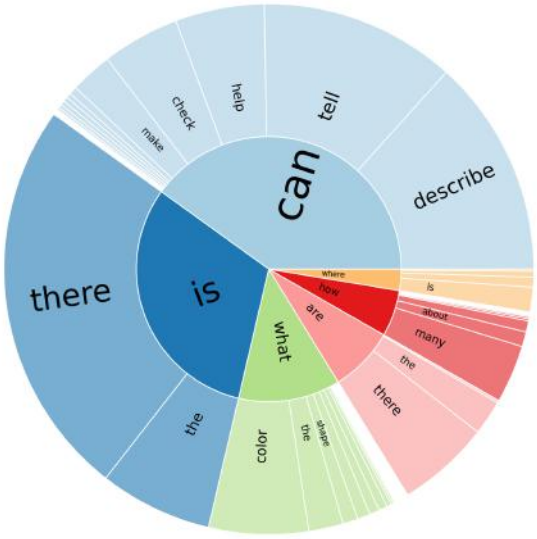


Figure A.12: Question types: 3RDialog.

D. Model Details

D.1. Prompts

The first portion of prompts sent into the LLM is a **system message**. It consists of two parts: a role prompt and a situation prompt. The role prompt is the same for all tasks:

> You are an AI visual assistant situated in a 3D scene. You can perceive (1) an ego-view image (accessible when necessary) and (2) the objects (including yourself) in the scene (always accessible). You should properly respond to the USER's instructions according to the given visual information.

The situation prompt begins with a common sentence:

> You are at a selected location in the 3D scene.

For SQA3D (Ma et al., 2023), the situation prompt is further extended with the situation description in the dataset. The situation prompt is only used jointly with the embodiment token to support tasks that require information about the embodiment. Details can be found in Appendix D.2.1.

Next are the **visual tokens**, including **2D image tokens** and **object-centric 3D tokens**. Each token sequence is interleaved

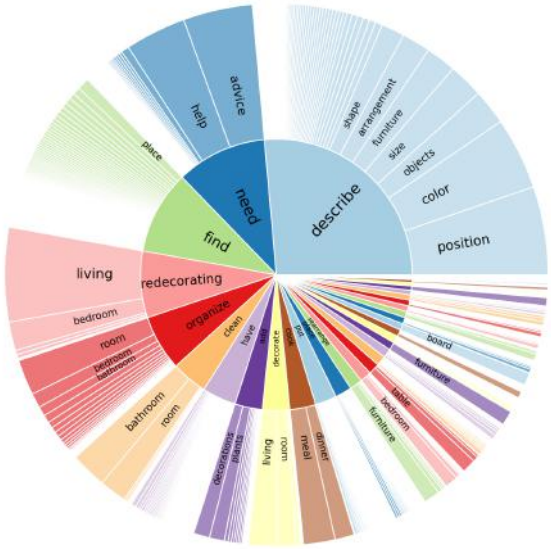


Figure A.13: Noun-verb pairs: 3RDialog instruction.

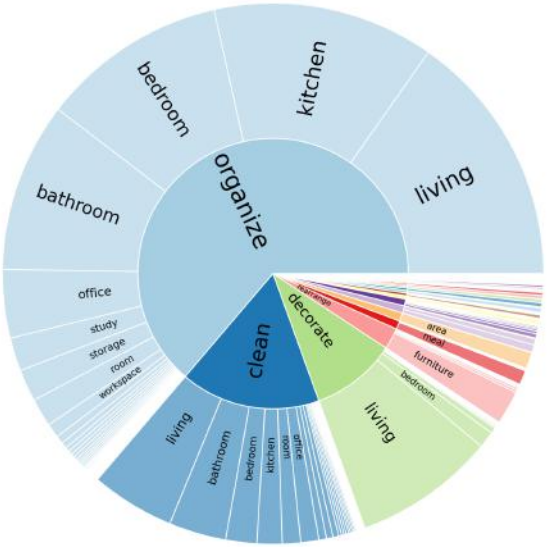


Figure A.14: Noun-verb pairs: 3RPlan instruction.

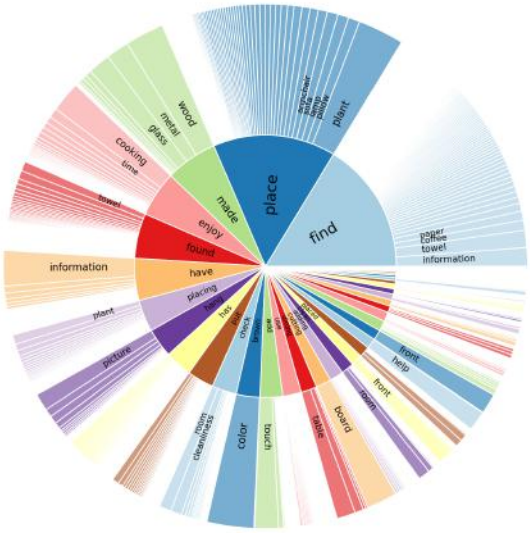


Figure A.15: Noun-verb pairs: 3RDialog response.

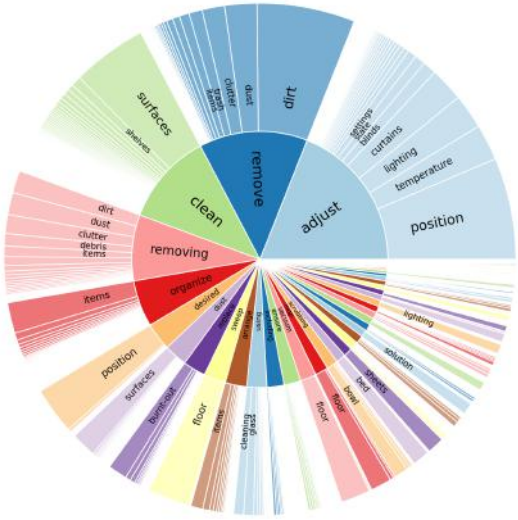


Figure A.16: Noun-verb pairs: 3RPlan response.

within text tokens and starts with a text prefix.

> Ego-view image: {IMAGE_TOKENS}
> Objects (including you) in the scene: {OBJECT_TOKENS}

The last portion of prompts is a **task-specific instruction**. For **object-level caption** and **object-in-the-scene caption**, we randomly chose one sentence from 151 sentences to be the instruction. Some examples can be found in Tab. A.8. For **scene-level caption**, we randomly choose one from 183 instructions. Examples can be found in Tab. A.9. For **3D question answering** task, we simply use the question as the instruction. The dialog history is used as the instruction for **3D dialogue** to provide continuity across multiple rounds of interactions. A planning instruction pool consisting of 202 instructions is introduced for **scene-aware task planning** and we randomly choose one from it as done in the caption tasks. Examples from the pool can be found in Tab. A.10. The chosen instruction is further followed by an instruction that specifies the task, *e.g.*, *set up a home office*.

Table A.8: **Examples from our object-level caption instruction set.**

"Produce a description for the object at the chosen spot in the 3D scene.",
"How would you depict the object located at the selected point in the 3D environment?",
"Formulate a description of the item at the picked position within the 3D scene.",
"How would you describe the entity at the designated location in the 3D backdrop?",
"Can you detail the object situated at the selected point in the 3D setting?",
"Compose a narrative for the object at the chosen locale within the 3D environment.",
"What does the object at the specified position in the 3D visualization look like?",
"Provide a description for the item located at the marked site in the 3D world.",
"How would you illustrate the object placed at the selected spot in the 3D landscape?",
"Craft a depiction of the object at the pinpointed location within the 3D territory.",
"What kind of object is illustrated at the identified site in the 3D tableau?",
"Develop a description of the object at the specified position in the 3D backdrop.",
"What is the entity's detail at the highlighted site in the 3D view?",
"Write up a description of the entity at the selected spot in the 3D realm.",
"What does the object look like at the pinpointed location in the 3D space?",
"Detail the entity located at the chosen position within the 3D scene.",
"Can you explain the essence of the object at the selected spot in the 3D zone?",

With past action tokens {PAST_ACTIONS} appended at the end, the instruction for **embodied navigation** is as follows, where {GOAL} stands for the goal specified by the target object name:

The task is navigation. Your goal is to find {GOAL} by moving around in the scene. Past actions: {PAST_ACTIONS}.

The instruction for **robotic manipulation** is similar to the one in **embodied navigation**. Here {GOAL} is the task description in CLIPort:

The task is manipulation. Your goal is to {GOAL}. Past actions: {PAST_ACTIONS}.

D.2. Feature Encoding

We have several modules to encode the multi-modal features.

- **Object-centric 3D token embedding.** The encoder for 3D object-centric point clouds is a PointNet++ (Qi et al., 2017) pre-trained on ScanNet (Dai et al., 2017) with object-classfication task. We sample 1024 points for every object as in (Chen et al., 2022). The architecture parameters all remain the same with (Chen et al., 2022). We freeze the PointNet++ for empirically better results.

- **Spatial Transformer (Chen et al., 2022).** Spatial Transformer is a modified transformer architecture that explicitly encodes spatial relations between object pairs. Specifically, consider the vanilla self-attention (Vaswani et al., 2017) mechanism which takes as input a feature matrix $X \in \mathbf{R}^{N \times d}$, where N stands for the number of tokens and d is the feature dimension. Vanilla self-attention first compute $Q = XW_Q, K = XW_K, V = XW_V$ from X using learnable projection matrices $W_Q, W_K, W_V \in \mathbf{R}^{d \times d_h}$ where d_h stands for the output feature dimension. Then the attention weight matrix is computed by $(\omega_{ij}^o)_{N \times N} = \Omega^o = softmax(\frac{QK^T}{\sqrt{d_h}})$ and finally used for re-weighting $\Omega^o V$. The intuition of Spatial Transformer is that we can re-scale the elements ω_{ij}^o in the weight matrix Ω^o.

  In the object-centric reasoning setting, the input feature matrix is $O \in \mathbf{R}^{N \times d}$. Consider an object pair (O_i, O_j) with their geometric centers c_i, c_j. Spatial Transformer (Chen et al., 2022) computes the Euclidean distance $d_{ij} = ||c_i - c_j||_2$ and the horizontal and vertical angles θ_h, θ_v of the line connecting c_i and c_j. The spatial feature between the two objects (O_i, O_j) is a 5-dimensional vector $f_{ij} = [d_{ij}, \sin(\theta_h), \cos(\theta_h), \sin(\theta_v), \cos(\theta_v)]$. To combine this feature with objects,

Table A.9: **Examples from our scene-level caption instruction set.**

"Describe this scene.",
"Generate a description of this scene.",
"Generate a caption of this scene.",
"Can you describe the scene?",
"Can you generate a description of the scene?",
"Can you generate a caption of the scene?",
"Summarize this scene.",
"Provide an outline of this 3D scene's characteristics.",
"How would you describe the 3D scene?",
"How would you summarize this scene?",
"Convey a summary of the 3D structure of this scene.",
"How would you interpret this 3D scene?",
"Offer a summary of the 3D scene.",
"Can you describe this scene in detail?",
"I'm interested in this scene, can you explain?",
"What is this scene made of?",
"Could you provide more info about this scene?",

Table A.10: **Examples from our planning instruction pool.**

"Plan for the task",
"Can you come up with a plan for this task",
"How can we do this task, provide a step-by-step plan",
"Draft a plan for completing this task",
"Detail a strategy for the task",
"What's the best plan for this task",
"Draw out a procedure for the task",
"Lay out the steps for this task",
"Could you devise a plan for the task",
"Show me a plan for this task",
"I need a plan for the task",
"Sketch a plan for the task at hand",
"Set up a plan for this",
"Recommend a plan for this task",
"Offer a strategy for this task",
"Design a blueprint for the task",
"Outline the approach for this task",

the spatial attention computes $\omega_{ij}^s = g_i f_{ij}$ where $g_i = W_S^T o_i$ is a 5-dimensional vector. The spatial attention further reweights the original self-attention weight matrix as

$$\omega_{ij} = \frac{\sigma(\omega_{ij}^s) exp(\omega_{ij}^o)}{\sum_{l=1}^{N} \sigma(\omega_{il}^s) exp(\omega_{il}^o)}.$$

Readers are referred to (Chen et al., 2022) for more details. In summary, Spatial Transformer explicitly computes pairwise spatial relations and fuses them with vanilla self-attention to provide better spatial reasoning ability. We use a three-layer Spatial Transformer with 8 heads to process the object-centric features produced by PointNet++ and output object tokens for LLM. For other settings, We follow all the default hyperparameters in (Chen et al., 2022).

- **2D token embedding.** We use OpenCLIP ConvNext-base model (Liu et al., 2022) pre-trained on LAION2B (Schuhmann

et al., 2022) to process the egocentric 2D image.

- **CLIP semantic guidance.** To inject more semantics into visual tokens, we use the text encoder from CLIP (Radford et al., 2021) to process the instruction tokens to obtain a global semantics feature. Next, we update the visual tokens via element-wise product between the CLIP semantics feature and each image & object token embedding.

D.2.1. Embodiment Encoding

In addition to the egocentric 2D input, we introduce an embodiment token to help LEO reason in an embodiment-aware fashion. We find it useful to use it together with the situation prompt and 2D egocentric input. Specifically, an embodiment token e is introduced in **embodied navigation**, **embodied reasoning**, and **object-in-the-scene caption** tasks. Specifically, e is a learnable embedding that will be inserted into the 3D object list.

So what does embodiment information mean in these tasks? In **embodied navigation**, it means the agent's position and orientation in the scene, which can be derived from a GPS and a compass sensor. The orientation of the agent is further represented by a rotation which is Fourier-embedded and mapped to a feature vector r by a linear layer. It is the same in **embodied reasoning** task. In the **object-in-the-scene caption** task, we assume the agent is situated at the location of the object that is being referred to. Therefore, embodiment information also means the location of the referred object. We obtain this location by randomly choosing a spot inside the referred object bounding box. To sum up, we could simply treat the embodiment token as a special *self object*, where its object embedding is learnable, and its location/orientation corresponds to the actual or assumed "agent".

After inserting the embodiment token, we obtain a new 3D object token list: $e, s_{\text{3D}}^{(1)}, s_{\text{3D}}^{(2)}, \ldots, s_{\text{3D}}^{(N)}$, where $s_{\text{3D}}^{(i)}, i \in \{1, 2, \ldots, N\}$ are 3D object token embeddings produced by PointNet++, along with location specified for each object (including the *self-object*). We can concatenate them together to get a feature matrix $O \in \mathbf{R}^{(N+1)\times d}$ and send them to the Spatial Transformer to explicitly fuse the spatial information of all the 3D objects and the self-object.

D.3. Action Tokenization

To empower LEO to exert control over an embodiment or a robot, we encode all actions within the context of Object Navigation (Ramrakhya et al., 2022) and CLIPort (Shridhar et al., 2021) tasks using the least frequently employed language tokens. Specifically, for the Object Navigation task, we allocate 4 tokens to represent actions of *move forward*, *turn right*, *turn left*, and *stop*. For the CLIPort task, we use a total of 516 tokens to discretize action poses, with 320 tokens dedicated to the x-axis pose bins, 160 tokens for the y-axis pose bins, and 36 tokens for the z-rotation bins.

D.4. LLM Hyperparameters

We set the maximum output length of our Vicuna-7B to be 256. The maximum context length is also set to 256 and if the length of the input is greater than 256, we truncate it to 256 by deleting tokens from the left (*i.e.*, only the rightmost 256 tokens are preserved). We set rank and α in LoRA (Hu et al., 2022) to be 16 and the dropout rate to be 0. LoRA is implemented for all the projection matrices in the LLM, *i.e.*, (W_q, W_k, W_v, W_o) in attention modules and $(W_{gate}, W_{up}, W_{down})$ in MLPs.

The hyperparameters for inference are listed in Tab. A.11.

E. Alignment Setup

The hyperparameters for 3D VL alignment are presented in Tab. A.12.

F. Instruction-tuning Setup

The hyperparameters for 3D VLA instruction tuning are presented in Tab. A.13.

Table A.11: Hyperparameters for LEO inference.

Hyperparameters	Value
Number of beams	5
Maximum output length	256
Minimum output length	1
Top p	0.9
Repetition penalty	3.0
Length penalty	1.0
Temperature	1.0

Table A.12: Hyperparameters for the alignment stage.

Hyperparameter	Value
Optimizer	AdamW
Weight decay	0.05
Betas	[0.9, 0.999]
Learning rate	3×10^{-4}
Warmup steps	400
Number of workers	4
Parallel strategy	DDP
Type of GPUs	NVIDIA A100
Number of GPUs	4
Accumulate gradient batches	5
Batch size per GPU (total)	4 (80)
Training precision	bfloat16
Gradient norm	5.0
Epochs	5

Table A.13: Hyperparameters for the instruction-tuning stage.

Hyperparameter	Value
Optimizer	AdamW
Weight decay	0.05
Betas	[0.9, 0.999]
Learning rate	3×10^{-5}
Warmup steps	400
Number of workers	4
Parallel strategy	DDP
Type of GPUs	NVIDIA A100
Number of GPUs	4
Accumulate gradient batches	5
Batch size per GPU (total)	4 (80)
Training precision	bfloat16
Gradient norm	5.0
Epochs	10

G. Ablation Details

G.1. Object-centric Mask

Ground truth *vs*. object proposals. As we adopt an object-centric 3D representation, the object-centric masks are necessary to segment the scene point cloud. For scenes that lack annotations of object-centric masks, we can utilize off-the-shelf detection or segmentation models to generate object proposals and thus obtain the masks. We compare the performances of LEO (*w/o Act*) between using ground-truth masks and Mask3D (Schult et al., 2022) proposals. The results in Tab. A.14 indicate that using Mask3D proposals leads to a moderate performance drop on Scan2Cap (mainly due to the IoU@0.5 metrics) and comparable performances on QA tasks.

Table A.14: Quantitative comparison between LEO (*w/o Act*) using ground-truth masks and Mask3D proposals. Metrics follow Tab. 4.

	Scan2Cap (val)					ScanQA (val)					SQA3D (test)
	C	B-4	M	R	Sim	C	B-4	M	R	EM@1	EM@1
w/o Act (Mask3D)	72.4	38.2	27.9	58.1	55.3	101.4	13.2	20.0	49.2	**24.5** (47.6)	50.0 (52.4)
w/o Act (GT)	**87.4**	**44.5**	**30.8**	**65.7**	**65.4**	**103.0**	**14.6**	**20.1**	**49.7**	24.3 (**48.5**)	50.0 (**52.5**)

Table A.15: Quantitative results of LEO equipped with LLMs at different scales. Metrics follow Tab. 7.

	ScanNet			3RScan		
	Scan2Cap	ScanQA	SQA3D	3RQA	3RDialog	3RPlan
w/o Act (OPT-1.3B)	64.6	20.3 (44.2)	45.5 (47.6)	50.0 (54.5)	71.1	78.3
w/o Act (Vicuna-7B)	**65.4**	**24.3** (48.5)	**50.0** (**52.5**)	51.9 (57.4)	**73.3**	**81.1**
w/o Act (Vicuna-13B)	65.2	23.4 (**48.9**)	49.7 (52.3)	**56.2** (**60.4**)	72.5	80.5

G.2. Model Ablation

LLM. Following the setting of LEO (*w/o Act*), we ablate the default LLM (Vicuna-7B) with OPT-1.3B (Zhang et al., 2022) and Vicuna-13B (Chiang et al., 2023), respectively. We report the evaluation results on ScanNet and 3RScan tasks in Tab. A.15. The results show a significant gap between OPT-1.3B and Vicuna-7B and comparable performances between Vicuna-7B and Vicuna-13B. This indicates the notable improvements when scaling from smaller LLM to 7B scale and the potential saturation if we continue to scale up, resembling the finding in Sec. 4.5.

Point cloud backbone. We have tried substituting PointNet++ (Qi et al., 2017) with Point-BERT (Yu et al., 2022) as the point cloud backbone. Specifically, we utilize the Point-BERT checkpoint from PointLLM (Xu et al., 2023), which has adapted Point-BERT to 6-channel (XYZRGB) input and learned a language-aligned representation for 3D objects. We have not observed notable difference between the performances of using Point-BERT and PointNet++ so we omit the results here.

G.3. Dialogue and Planning Data

To evaluate *w/o Dialg*, we design an evaluation set with three types of questions: 1) **Answerable**: general questions that can be answered based on the given 3D scenes; 2) **Unanswerable**: questions that cannot be answered given the 3D scenes due to a lack of information, *e.g.*, "Tell me about the elephant in the room"; 3) **NLP**: questions that solely examine the language functionality of LEO in term of factual knowledge, reasoning, and text coherence. We collect 30 representative questions for each subset and generate LEO's responses for each question. We then ask humans to choose their preferred responses between *w/o Dialg* and *w/ Dialg* Based on the human preferences, we evaluate the two models with TrueSkill (Graepel et al., 2007), which is an algorithm that quantifies players' rating scores by Bayesian inference. The scores are estimated by Gaussian distribution and expressed as $\mu \pm \sigma$.

G.4. Data Balancing

To investigate the hallucination problem, we collect 150 questions querying object existence on 3RScan and ScanNet respectively. We split three subsets according to the category of queried object. The queried object can exist in the given scene (Yes), exist in other scenes instead of the given scene (No-1), or not exist in all the scenes (No-2). Each subset comprises 50 questions. We merge No-1 and No-2 when reporting the exact-match accuracy, as shown in Tab. 9.

H. Evaluation Details

H.1. 3D Question Answering

Rationality of QA evaluation protocol. We argue that exact match (EM), as a conventional metric for 3D QA, is unsuitable for evaluating the open-ended answer generated by LLMs. For example, given the question "*On what side of the towel is a bathroom curtain*?" with ground-truth answer "*left side of towel*", it is never wrong to answer "left". However, this will be deemed incorrect if we adopt the strict exact match protocol. Such a misjudgment is quite likely to occur when evaluating the answers from LLMs. By contrast, the classifier heads for QA (*e.g.*, MCAN) are less affected because they collect all possible answers in advance to formulate the QA as a close-set classification problem. Hence, we refine the strict

Table A.16: Examples from ScanQA validation set manifest the rationality of our refined exact match protocol.

Question	Ground-truth answer	Predicted answer	Strict EM	Refined EM
What color is the chair in the kitchen?	dark brown	brown	✗	✓(case 2)
What is under the long kitchen counter?	kitchen cabinets	brown rectangular kitchen cabinets	✗	✓(case 2)
What type of refrigerator is on the right of a kitchen counter?	stainless steel refrigerator	stainless steel	✗	✓(case 2)
Where is the beige wooden desk placed?	up against wall	against wall	✗	✓(case 2)
What color does the sofa look?	it looks black	black	✗	✓(case 2)
Where is the black office chair located?	in front of desks	in front of desk	✗	✓(case 2)
What is in the corner by windows?	book shelf	bookshelf	✗	✓(case 2)
Where is the chair pulled into?	table	under table	✗	✓(case 3)
How many chairs are to the left of the table?	4	4 chairs	✗	✓(case 3)
What objects are sitting on the black couch?	pillow	pillows	✗	✓(case 3)
Where are the two different size tables located in room?	in center	in center of room	✗	✓(case 3)
Where is the laptop located?	desk	on desk	✗	✓(case 3)
Where is the soap dispenser mounted	above sink	on wall above sink	✗	✓(case 3)

exact match protocol as follows.

```
"""
code for QA protocols
pred: str
gts: List[str]
"""

def strict_em(pred, gts):
    for gt in gts:
        if pred == gt:
            # case 1
            return True


def refined_em(pred, gts):
    for gt in gts:
        if pred == gt:
            # case 1
            return True
        elif ''.join(pred.split()) in ''.join(gt.split()):
            # case 2
            return True
        elif ''.join(gt.split()) in ''.join(pred.split()):
            # case 3
            return True
    return False
```

In a nutshell, we squeeze the `pred` and `gt`, and then check whether one is a subset of the other. To justify our refined exact match protocol, in Tab. A.16 we provide some representative examples in the ScanQA validation set. Despite the improvements, we speculate such a simple refinement is still insufficient for a sound evaluation metric considering the flexibility of human language.

H.2. Embodied Navigation

To construct our training set, we adopt all 57 scenes in the MP3D `ObjNav` training split (Savva et al., 2019; Ramrakhya et al., 2022) and generate ~60K shortest-path navigation episodes. The evaluation is conducted on the original validation split of the MP3D `ObjNav` task and the newly introduced HM3D `ObjNav` task (Ramakrishnan et al., 2021).

In contrast to most `ObjNav` agents that utilize recurrence through either RNN (Ramrakhya et al., 2022) or DT-style Transformer (Suglia et al., 2021), LEO only employs a simplistic feed-forward policy, *i.e.*, the Transformer in LEO only takes in the instruction, current state (2D and 3D observation), and past 4 actions, and predicts the next action, similar to RT-2 (Brohan et al., 2023). Therefore, the only information relayed from the past is past actions. The absence of recurrence in LEO's acting policy is indeed the result of a trade-off between better performances and training efficiency. We will commit to exploring the possibility of looping in more sophisticated policy architectures (*e.g.*, recurrence) in future work.

Table A.17: Quantitative comparison between LEO pretrained on the generated data before/after refinement. Metrics follow Tab. 4.

	Scan2Cap (val)					ScanQA (val)					SQA3D (test)
	C	B-4	M	R	Sim	C	B-4	M	R	EM@1	EM@1
Before refinement	84.1	**45.8**	30.9	66.1	65.3	99.4	12.6	19.4	48.6	24.5 (49.1)	48.2 (50.5)
After refinement	**87.1**	45.2	**31.1**	66.1	**65.7**	**105.7**	**14.9**	**20.5**	**50.7**	**24.7 (49.8)**	**52.4 (55.0)**

Table A.18: Quantitative comparison between LEO trained on the LL3DA data and our data. Metrics follow Tab. 4.

	Scan2Cap (val)					ScanQA (val)				
	C	B-4	M	R	Sim	C	B-4	M	R	EM@1
LL3DA data	73.9	43.5	30.2	65.0	63.4	99.7	**14.8**	19.7	47.8	22.9 (46.4)
Our data	**86.4**	**44.4**	**30.9**	**65.8**	**65.6**	**104.9**	13.8	**20.4**	**50.3**	**24.5 (49.2)**

Table A.19: Quantitative comparison between LL3DA and LEO when both trained on LL3DA data. Metrics follow Tab. 4.

	Scan2Cap (val)				Nr3D (val)				ScanQA (val)			
	C	B-4	M	R	C	B-4	M	R	C	B-4	M	R
LL3DA	63.0	36.0	25.7	54.7	**23.9**	**13.4**	22.3	45.8	75.7	13.3	15.4	37.0
LEO	**64.9**	**37.2**	**27.4**	**57.5**	22.1	10.9	**22.9**	**46.3**	**99.2**	**14.9**	**19.4**	**47.3**

I. Additional Results

I.1. Impact of Data Refinement

Settings. We investigate the impact of data refinement by comparing the downstream performances between pretraining on the generated data before/after refinement. Specifically, since our generated data (where the refinement occurs) pertains to 3RScan scenes, we first pretrain the LEO after the alignment stage on a mix of 3RScan datasets, and then train on a mix of ScanNet datasets (Scan2Cap, ScanQA, and SQA), where we report the quantitative results as downstream performances.

The results in Tab. A.17 demonstrate that data refinement elicits consistent improvements. In particular, data refinement primarily benefits reasoning (QA) tasks, probably because the refinement operation mainly concerns QA and dialogue data.

I.2. Data Comparison

Settings. We collect the training data of LL3DA (Chen et al., 2024) to train LEO and compare the quantitative results with LEO trained with our original data to showcase the impact of training data. We report the performances on Scan2Cap and ScanQA, where their data overlaps ours.

The results in Tab. A.18 exhibit a consistent performance gap between training on LL3DA data and our original data, underscoring the advantage of our collected training data.

I.3. Model Comparison

Settings. LEO adopts an object-centric 3D representation to encode 3D scenes, which is a novel approach compared with recent works. For example, 3D-LLM (Hong et al., 2023) leverages 2D foundation models to obtain dense semantic features and lift them to 3D space, and LL3DA (Chen et al., 2024) adopts scene-level encoding. They both use learnable queries to extract 3D features. Here we investigate the influence of model design with the same training data. For a fair comparison, we use Mask3D (Schult et al., 2022) object proposals instead of ground-truth masks for the evaluation results of LEO.

LL3DA *vs*. LEO. We train LEO on the LL3DA training data and compare the performances with LL3DA generalist results (without task-specific fine-tuning). From the results in Tab. A.19, we highlight two takeaways: 1) with the same training data, LEO outperforms LL3DA on most metrics; 2) the gap between LL3DA and LEO is significant on ScanQA, which indicates a major advantage of object-centric 3D representation lies in handling the reasoning task.

3D-LLM *vs*. LEO. As LL3DA collects a subset (ScanNet part) of 3D-LLM training data, we leverage this subset to pretrain LEO and compare the downstream performances with 3D-LLM. In contrast to the task-specific fine-tuning results of 3D-LLM, we report LEO's evaluation results after instruction tuning without task-specific fine-tuning. The results in

Table A.20: Quantitative comparison between 3D-LLM and LEO when both trained on 3D-LLM data. Metrics follow Tab. 4.

	ScanQA (val)					SQA3D (test)
	C	B-4	M	R	EM@1	EM@1
3D-LLM	74.5	12.9	15.1	37.5	21.2	49.8
LEO	**97.4**	**14.6**	**19.1**	**46.8**	**23.2** (45.4)	**50.6** (52.9)

Table A.21: **Results on object navigation with OOD objects and human demonstrations.** Note that the baseline Habitat-web is unable to generalize to MP3D-unseen as it uses categorical embedding rather than natural language to represent object goals.

	MP3D-seen		MP3D-unseen	
	Success(↑)	SPL(↑)	Success(↑)	SPL(↑)
Habitat-web (shortest)	4.4	2.2	-	-
Habitat-web (70k demo)	**35.4**	10.2	-	-
LEO (shortest, w/o 2D)	7.8	4.6	-	-
LEO (shortest, w/o 3D)	8.6	6.8	-	-
LEO (shortest)	23.1	**15.2**	**11.1**	**9.6**
LEO (70k demo)	7.1	5.3	8.9	8.6

Tab. A.20 show that LEO consistently outperforms 3D-LLM when adopting the same training data. Notably, the magnitude of this subset is much smaller than their original training data, which further underscores the efficiency of our model.

I.4. Embodied Acting

Quantitative results of `ObjNav`. We provide additional results of LEO 1) generalizing to unseen objects on MP3D (below is a list of the objects used during training (seen) and for OOD evaluation (unseen)), 2) learning with 70K human demonstrations provided by Habitat-web (Ramrakhya et al., 2022) instead of shortest path, and 3) learning without one modality (full vs. w/o 3D vs. w/o 2D). Evaluation results are shown in Tab. A.21. Note that the baseline Habitat-web is unable to generalize to novel objects as it uses categorical embedding rather than natural language to represent object goals.

```
# Objects (seen)
"gym_equipment", "tv_monitor", "picture", "counter", "chair", "cabinet",
"table", "stool", "plant", "towel", "sofa", "cushion", "sink", "fireplace",
"toilet", "seating", "chest_of_drawers", "bed", "shower", "bathtub",
"clothes"

# Objects (unseen)
"shelf", "pillow", "lamp", "box", "desk", "refrigerator", "vase", "armchair"
```

The results show that LEO can generalize to novel objects. On the other hand, human demonstrations include more explorations, compared with shortest-path data. Therefore, it will be much harder for agents without a recurrent module (*e.g.*, LEO) to learn from human demonstrations (see Appendix H.2), leading to significantly weaker performances. Finally, more modalities (2D + 3D) are indeed crucial to the success of mastering object navigation skills.

Qualitative results. We provide more qualitative results of robotic manipulation and embodied navigation in the supplementary video.

I.5. Scan2Cap

We provide additional qualitative results on Scan2Cap validation set in Tab. A.22. The results show that LEO can correctly refer to the queried object and provide accurate descriptions, including spatial relationships with other objects. However, LEO's responses are confined to simple formats that lack diversity. How to unlock more flexible responses while maintaining accuracy can be a direction for future research.

I.6. ScanQA

We provide additional qualitative results on ScanQA validation set in Tab. A.23 and categorize the responses into several types:

- *Wrong*. The response is inaccurate and deemed wrong.
- *Wrong but reasonable*. The response is deemed wrong but is reasonable to some extent, probably due to ambiguities in the scene. Consider the second case in Tab. A.23. There are many objects such as a coat rack, a coat, and a mini fridge-shaped cabinet on the right side of the organizer. Though LEO's response "mini fridge" does not match the ground truth "coat rack", it is consistent with the 3D scene layout.
- *Wrong but accurate*. The response is accurate according to the scene but is deemed wrong due to imperfect ground truth annotations.
- *Correct*. The response is accurate and deemed correct.
- *Correct and more accurate*. The response is more accurate than the ground truth annotations.

I.7. SQA3D

We provide additional qualitative results on SQA3D test set in Tab. A.24 and follow the aforementioned response types. The embodied reasoning in SQA3D requires the understanding of not only the scene but also the situation of embodiment. In Tab. A.24, answering "What am I sitting at?" necessitates that LEO accurately identifies the objects at its current location. And the response to "How many beds are in front of me?" indicates that LEO can reason based on the understanding of its orientation.

Table A.22: **Additional qualitative results on Scan2Cap validation set**.

Scene	Instruction	Response	Ground Truth
	Develop a description of the object at the specified position in the 3D backdrop.	This is a black keyboard. It is in front of two monitors.	The keyboard sits atop the workstation desk. It is located in front of the left-side monitor.
	Focus on the object at the given 3D location and describe it.	This is a white pillow. It is on the couch.	It is a small pillow located on the couch. You can notice it directly on your left when walking through the door into the room.
	Provide a clearer description of the object at that specific 3D spot.	This is a white cabinet. It is to the right of a couch.	This is a white ottoman. It is located to the right of the black couch.
	What does the object at the specified position in the 3D visualization look like?	There is a rectangular brown door. It is next to a couch.	There is a door in the middle of the northern wall next to a metal cabinet. The door is shaped like a brown rectangle and as a dark yellow colored doorknob.

Table A.23: **Additional qualitative results on ScanQA validation set**.

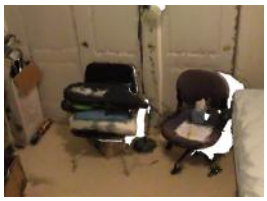
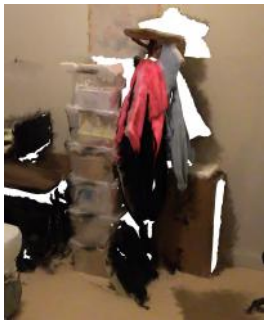
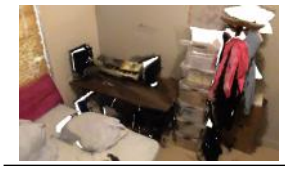
Scene	Question	Response	Ground Truth	Response Type
	What is placed on the black chair out of the three chairs present in the room?	pillow	tan jacket, guitar box like object	*Wrong*
	What is next to the organizer shelf?	mini fridge	coat rack	*Wrong but reasonable*
	Where is the dresser located?	to right of door	underneath television next to black guitar case and green guitar case, under tv set next to guitar cases	*Wrong but accurate*
	What kind of desk is the electric piano on?	wooden desk	wooden desk	*Correct*
	How many monitor sits on top of the desk?	2	2	*Correct*
	What is on the left side of the brown desk?	black computer monitor	monitor / computer monitor	*Correct and more accurate*

Table A.24: **Additional qualitative results on SQA3D test set**.

Scene	Question	Response	Ground Truth	Response Type
	What is the only orange object on my left side?	backpack	jacket	*Wrong*
	Is there anything on the desk?	yes	laptop	*Wrong but accurate*
	What type of bed is the one on the right?	loft bed	loft bed	*Correct*
	What am I sitting at?	desk	desk	*Correct*
	How many beds are in front of me?	one	one	*Correct*

Table A.25: **Examples from our datasets**. We provide the tasks, 2D images (optional), 3D point clouds, and instructions as the input of LEO, as well as the expected responses of LEO.

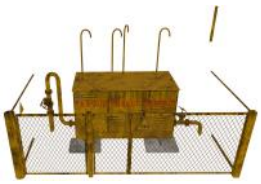
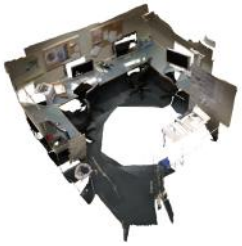
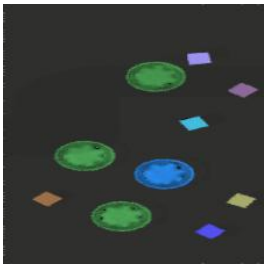
Dataset & Task	2D & 3D input	Instruction	Response
LEO-align object captioning		USER: Explain the object at the pinpointed location in the 3D setting. ASSISTANT:	a yellow metal cage with a fence around it.
LEO-align object referring		USER: Give me a snapshot of the object at the chosen 3D location. ASSISTANT:	find the office chair that is near the copier
LEO-align scene captioning		USER: Relay a brief summary of this scene's elements. ASSISTANT:	In this scene, there is a gray flat picture hanging on the wall, standing in front of a windowsill. There is also a rectangular white padded pillow lying on the bed, close to a big rectangular white padded pillow. The room has a white architectural wall and a flat overhead ceiling. The bed is tidy and white, located in front of a narrow tall commode. The curtains are hanging on the wall. The room has a calm and comfortable atmosphere, suitable for relaxation and sleep.

Table A.26: **Examples from our datasets**. Continued.

LEO-instruct 3D captioning		USER: What does the object at the specified position in the 3D visualization look like? ASSISTANT:	there is a door in the middle of the northern wall next to a metal cabinet. the door is shaped like a brown rectangle and as a dark yellow colored doorknob.
LEO-instruct 3D QA		USER: What is in the right corner of room by curtains? ASSISTANT:	brown cabinet with tv sitting in it
LEO-instruct 3D dialogue		USER: I'm redecorating my living room. Can you give me some suggestions on how to arrange the furniture? ASSISTANT: Of course! I'd be happy to help. What furniture do you have in your living room? USER: I have a sofa, a table, a chair, and a couch table. ASSISTANT:	Great! Here's a suggestion: you can place the sofa against the wall, with the table in front of it. The chair can be placed to the right of the table, and the couch table can be placed to the right of the chair. This arrangement creates a cozy seating area.
LEO-instruct task planning		USER: Develop a plan for this: Arrange and embellish the family room. ASSISTANT:	1. Arrange and fluff the cushions and pillows on the armchair. 2. Place decorations, such as figurines or vases, on the cabinet or shelf. 3. Place hygiene products, such as wipes or lotions, on the commode. 4. Turn on the lights and adjust their brightness if needed. 5. Arrange diapers in a designated storage area. 6. Place a blanket or pillow on the armchair for added comfort. 7. Adjust the festoon or other decorative elements if desired.

Table A.27: **Examples from our datasets**. Continued.

LEO-instruct embodied navigation	 	USER: The task is navigation. Your goal is to find counter by moving around in the scene. Past actions: <31999> <31999> <31999> <31999>. ASSISTANT:	<31996>
LEO-instruct robotic manipulation	 	USER: The task is manipulation. Your goal is to put the blue blocks in a green bowl. Past actions: <31991> <31671> <31511> <31991> <31671> <31511> <31991> <31671> <31511> <31991> <31671> <31511> <31991> <31671> <31511> <31991> <31671> <31511> <31991> <31671> <31511> <31991> <31671> <31511>. ASSISTANT:	<31748> <31644> <31511> <31736> <31595> <31500>