RoomTour3D: Geometry-Aware Video-Instruction Tuning for Embodied Navigation

Mingfei Han[1,3], Liang Ma[1], Kamila Zhumakhanova[1], Ekaterina Radionova[1], Jingyi Zhang[2]
Xiaojun Chang[1,4], Xiaodan Liang[1], Ivan Laptev[1]
[1]Department of Computer Vision, MBZUAI [2]Shenzhen Campus of Sun Yat-Sen University
[3]ReLER Lab, AAII, UTS [4]University of Science and Technology of China
https://roomtour3d.github.io

Abstract

Vision-and-Language Navigation (VLN) suffers from the limited diversity and scale of training data, primarily constrained by the manual curation of existing simulators. To address this, we introduce RoomTour3D, a video-instruction dataset derived from web-based room tour videos that capture real-world indoor spaces and human walking demonstrations. Unlike existing VLN datasets, RoomTour3D leverages the scale and diversity of online videos to generate open-ended human walking trajectories and open-world navigable instructions. To compensate for the lack of navigation data in online videos, we perform 3D reconstruction and obtain 3D trajectories of walking paths augmented with additional information on the room types, object locations and 3D shape of surrounding scenes. Our dataset includes ∼100K open-ended description-enriched trajectories with ∼200K instructions, and 17K action-enriched trajectories from 1847 room tour environments. We demonstrate experimentally that RoomTour3D enables significant improvements across multiple VLN tasks including CVDN, SOON, R2R, and REVERIE. Moreover, RoomTour3D facilitates the development of trainable zero-shot VLN agents, showcasing the potential and challenges of advancing towards open-world navigation.

1. Introduction

Over the past years, Vision-and-Language Navigation (VLN) [2, 25, 28, 36, 46, 58] has largely relied on human-designed simulators and annotated trajectories. R2R [2] established a benchmark for language-guided navigation in simulated indoor settings, while CVDN [55], REVERIE [46], and SOON [71] expanded VLN to dialogue-based and object-focused tasks. However, these manually curated simulations lack scene diversity and fail to capture real-world complexity.

To address limited diversity, recent methods propose the use of richer and more varied training data. AirBERT [15] combines discrete Airbnb images for panoramic views, which lack consistency and naturalistic context of an indoor scene. ScaleVLN [59] utilizes laboriously curated 3D scenes [49, 63], but suffers from reconstruction quality and scalability. More recently, YTB-VLN [38] attempts to use video frames to compose panoramic views and organize instructions with predefined templates, yet overlooks object variety and geometry structure. NaVid [65] constructs sequential single-view trajectories from MatterPort3D [4] and R2R [2] annotations, paired with general video data to train a sim-to-real agent. None of these approaches simultaneously achieves scalability in scene diversity, openness in object variety, or comprehensive geo-perception in spatial representations, each of which is critical to training effective and open-world navigation agents.

To address the challenge, we introduce **RoomTour3D**, a novel dataset that provides a geometry-aware, spatially enriched training environment for VLN agents. Built upon easily accessible room tour videos from the Internet, RoomTour3D captures continuous movement through real estates with a hand-held camera from a first-person perspective. Each frame presents a realistic, agent-centric view and showcases a rich array of indoor items. The continuous flow of these frames captures multiple views of the environment, presenting diverse room layouts and inherently embedding the geometric properties of the spaces. To unleash the power of these videos, we propose an automatic and extendable pipeline to obtain open-ended geometry-aware human walking trajectories, spatially contextualized textual instructions using open vocabularies.

To better model the navigation scenario, we take advantage of the continuous walk-through trajectories and densely sample frames from room tour videos. Then, we use COLMAP [50, 51] to reconstruct 3D scenes of real-estates to obtain the geometric information. With access to camera locations and orientations, we sample "decision-making" frames at points of maximal yaw rotation and sequence further frames every ∼1.5 meters to finalize trajectories. Additionally, our pipeline incorporates extensive annotations by employing RAM [66] for object tagging, Grounding-DINO [42] for precise localization, and Depth-Anything [64] to assess the relative distances between objects and the camera. To integrate knowledge of object variety, geometric awareness, and human walking preferences into model training, we employ GPT-4 [44] to generate navigation instructions for both summarization and task-specific navigation tasks.

Our RoomTour3D is an ongoing effort to create a comprehensive database derived from room tour videos and enriched by human-living knowledge. Currently, the dataset includes ∼100K open-ended trajectories with ∼200K descriptions, and ∼17K geometry-aware trajectories with navigable actions from 1847 homes. Moreover, we are releasing intermediate products such as object tags, bounding boxes, depth maps, room locations, and the necessary code and prompts used to generate instructions. To validate the robustness, we conducted experiments with NaviLLM [68], a generalist model based on Large Language Model (LLM), to train a unified multi-task navigation agent. Integrating our data into training simultaneously enhanced baseline performances such as CVDN, SOON, R2R and REVERIE with improvement exceeding 6%, achieving an outstanding 9.8% boost on SOON and setting new state-of-the-art (SOTA) results. Furthermore, our enriched action-instruction data enables the training of an end-to-end zero-shot navigation agent, advancing towards open-world embodied navigation.

In this work, we make the following key contributions:

- **Video Collection for Complex Environments:** We curate a novel dataset of diverse videos tailored for navigation tasks, distinguishing it from existing datasets such as YTB-VLN [38]. Our dataset features longer videos, enables representation of more complex environments, and exhibits fewer shot changes to ensure continuity and contextual consistency.
- **Automated Pre-processing of Videos:** We propose a pipeline to automatically extract geometry-aware navigation instructions, aligning spatial understanding with navigation goals. Additionally, we generate open-vocabulary instructions for diverse, open-ended trajectories to enhance real-world applicability.
- **Demonstrating Data Effectiveness:** Through extensive experiments and ablation studies, we demonstrate that our dataset significantly improves the performance of state-of-the-art models.

2. Related Work

2.1. Vision-and-Language Navigation

Learning to navigate unseen indoor environments with natural language instructions is vital for enabling embodied agents to assist humans. Various scenarios have been explored, including fine-grained instruction following (R2R [2]) and dialogue-based navigation (CVDN [55]), object localization from instructions (SOON [71], REVERIE [46]), and embodied question answering through active 3D exploration [10, 61]. While substantial work focuses on task-specific models [1, 6, 7, 14, 20–24, 29, 34, 41, 43, 47, 48, 56, 67, 71], they often lack generalization across tasks. In this context, NaviLLM [68] introduces an embodied generalist model that simultaneously addresses multiple tasks through a single framework, demonstrating strong generalization ability.

2.2. Data-Centric Methods for VLN

The scarcity of VLN training data remains a critical issue and results in poor generalization of VLN agents to unseen environments. Most of existing VLN datasets such as R2R [2], RxR [28], CVDN [55] and SOON [71] are produced in simulators, which constrains data scalability due to the high labor costs involved. To tackle the problem, data augmentation [12, 13, 27, 31, 32, 39, 53] and self-exploration in simulator environments [35, 57] have been investigated.

VLN-BERT [21] and AirBERT [15] attempt to use web-based image-caption pairs for pre-training, however, resulting trajectories often fail to mimic realistic navigation. Similarly, automatic dataset generation pipelines [8, 26], including ScaleVLN [59], rely on manually curated 3D scenes or synthetic environments, which are costly to produce and lack the photorealism needed for robust real-world generalization. PanoGen [30] enhances VLN training by generating diverse text-conditioned panoramic environments using text-to-image diffusion models and recursive outpainting. While this approach addresses the scarcity of training environments, it relies on synthetic panoramas and may not generalize well to real environments. YTB-VLN [38] advances scalability by leveraging YouTube room tour videos to generate path-instruction pairs but omits explicit path geometry, essential for robotic navigation.

In our work, we address the limitations by designing RoomTour3D with properties: (i) free-form and open-vocabulary path annotations instead of template-based instructions, (ii) extraction of open-ended trajectories from sequential video clips, and (iii) inclusion of turning points and spatially close frames as navigable candidate actions, moving beyond panoramic nodes. Furthermore, our approach integrates 3D reconstruction of indoor videos to retrieve trajectory geometry and employs an LLM to generate detailed, object-aware instructions with enhanced spatial understanding.

2.3. Zero-shot Navigation

Given the substantial semantic variations in complex real-world scenarios, fully-supervised VLN models often strug-

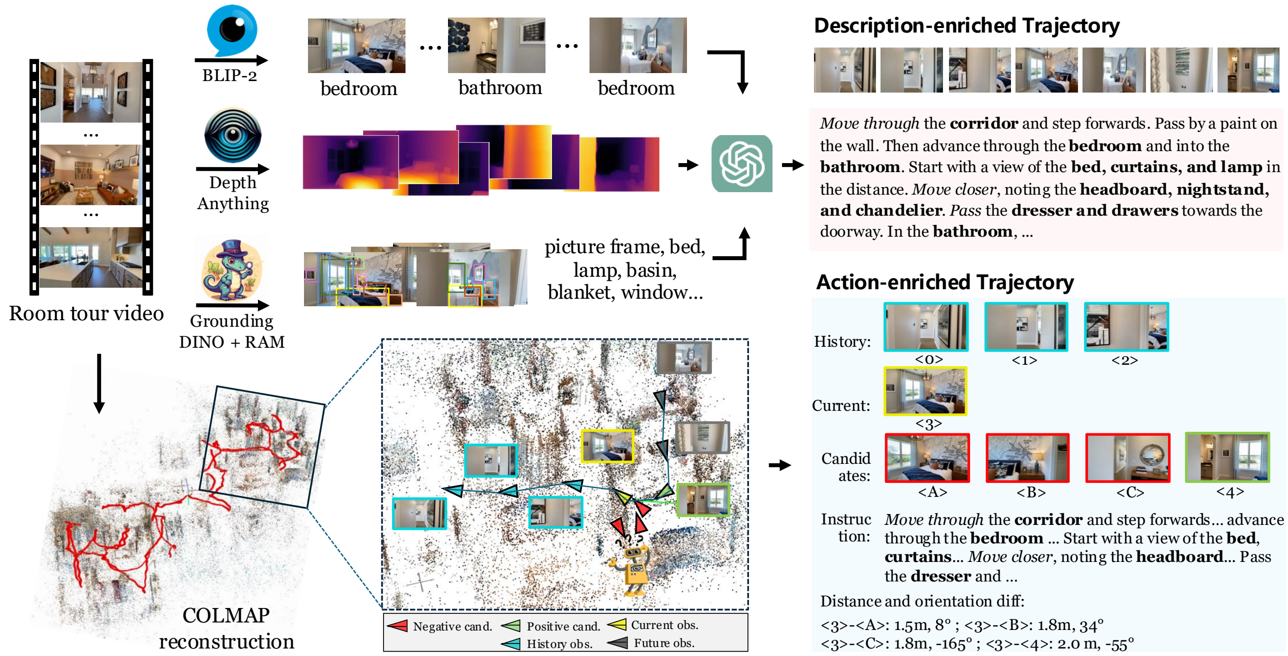

Room tour video
BLIP-2
Depth Anything
Grounding DINO + RAM
bedroom
bathroom
bedroom
picture frame, bed, lamp, basin, blanket, window...
COLMAP reconstruction
Negative cand.
Positive cand.
Current obs.
History obs.
Future obs.
Description-enriched Trajectory
Move through the corridor and step forwards. Pass by a paint on the wall. Then advance through the bedroom and into the bathroom. Start with a view of the bed, curtains, and lamp in the distance. Move closer, noting the headboard, nightstand, and chandelier. Pass the dresser and drawers towards the doorway. In the bathroom, ...
Action-enriched Trajectory
History:
<0>
<1>
<2>
Current:
<3>
Candidates:
<A>
<B>
<C>
<4>
Instruction:
Move through the corridor and step forwards... advance through the bedroom ... Start with a view of the bed, curtains... Move closer, noting the headboard... Pass the dresser and ...
Distance and orientation diff:
<3>-<A>: 1.5m, 8° ; <3>-<B>: 1.8m, 34°
<3>-<C>: 1.8m, -165° ; <3>-<4>: 2.0 m, -55°

Figure 1. Overview of our RoomTour3D data generation. Starting from a room tour video, we first apply BLIP-2 [33] on frame sequence to predict the room locations. Next, we use RAM [66] and Grounding-DINO [42] to identify objects within the frames and employ Depth-Anything [64] for depth prediction. Subsequently, COLMAP is used to reconstruct the 3D scene with complete geometry information, and we sample human walking trajectories from the continuous frames. The trajectory captures open-world objects, their positions, and depths relative to the camera. Finally, we use advanced LLM, *i.e.*, GPT-4 to generate the free-form descriptions for pretraining, namely description-enriched trajectories. Specifically, for the trajectory shown in the figure, which involves instant turning points, we specially treat <0> to <6> as walking trajectory, <A> <B> and <C> as side-watching points and use them as negative candidates for navigation finetuning task, namely action-enriched trajectories. For more details, please refer to Section 3.

gle to generalize across diverse navigation scenes. Zero-shot VLN has thus gained attention as it eliminates prior knowledge of environments and instructions, effectively mitigating environmental biases.

Commercial model based methods utilize advanced LLMs and robust frameworks for seamless solutions. MapGPT [5] incorporates a map-based prompting system with global spatial reasoning and adaptive path planning. DiscussNav [43] employs a multi-expert framework where LLMs specialize in subtasks like instruction analysis and vision perception. NavGPT [69] focuses on explicit reasoning by combining commonsense reasoning with visual observations. As to non-commercial methods, LangNav [45] uses language as the primary perceptual space, while Nav-CoT [37] introduces parameter-efficient training to allow LLMs to autonomously reason and act.

We show that usage of our action-enriched data for navigation tuning results in superior zero-shot performance over all non-commercial methods and reaches comparable results to commercial approaches based on GPT-3.5.

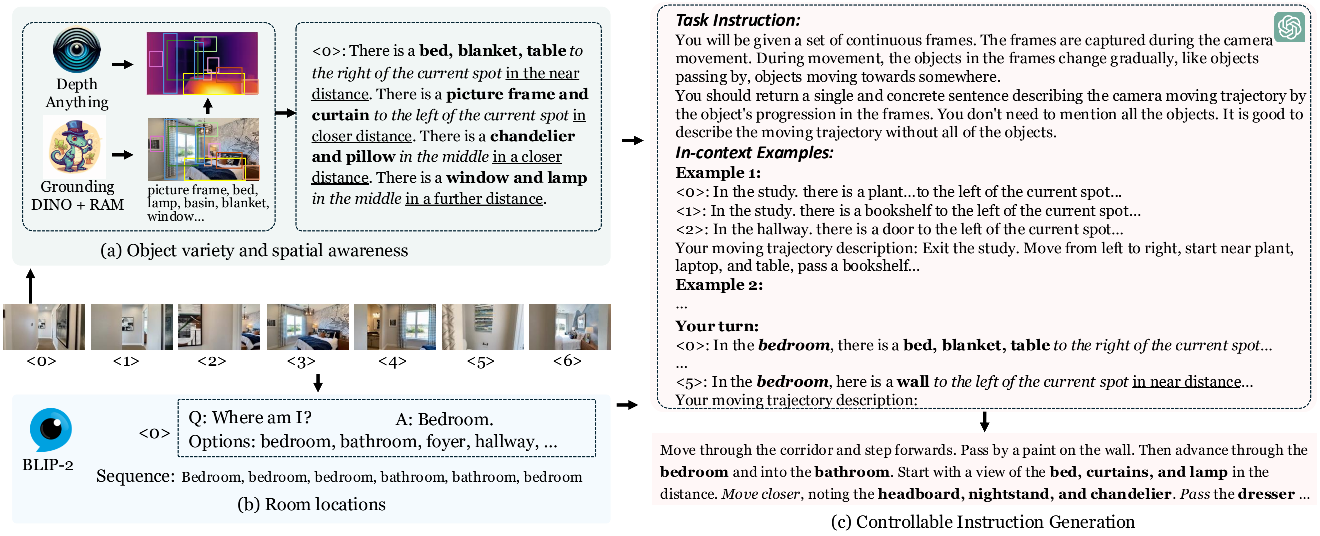
3. RoomTour3D

In this section, we present the automatic data curation pipeline of RoomTour3D. We detail the process of annotations, from sampling open-ended human walking trajectories to generating corresponding descriptions with open-world object variety and spatial awareness. Enabled by reconstructed 3D scenes, we further sample navigable trajectories with actions. The overall pipeline of our data generation is illustrated in Figure 1. Please refer to Appendix A for details about video collection.

3.1. Description-Enriched Trajectories

In this subsection, we detail the process of generating controllable descriptions for open-ended trajectories. We start by generating human-walking trajectories by uniformly sampling frames at a rate of one frame every two seconds, which aligns with the average human walking speed of 1.42 meters per second [62], typically slower in indoor environments. Subsequently, to annotate these trajectories, as shown in Figure 2, we employ expert models such as BLIP-2 [33], RAM [66], Grounding-DINO [42], and Depth-Anything [64] to gather extensive information on object variety, spatial positions, and depth measurements. Finally, we integrate this information into GPT4 [44] to generate detailed and coherent traje.

Object Variety and Spatial Awareness. In order to harness object variety and enable spatial awareness, we com-



<0>: There is a **bed, blanket, table** *to the right of the current spot* in the near distance. There is a **picture frame and curtain** *to the left of the current spot* in closer distance. There is a **chandelier and pillow** *in the middle* in a closer distance. There is a **window and lamp** *in the middle* in a further distance.

(a) Object variety and spatial awareness

Q: Where am I? A: Bedroom.
Options: bedroom, bathroom, foyer, hallway, ...

Sequence: Bedroom, bedroom, bedroom, bathroom, bathroom, bedroom

(b) Room locations

Task Instruction:
You will be given a set of continuous frames. The frames are captured during the camera movement. During movement, the objects in the frames change gradually, like objects passing by, objects moving towards somewhere.
You should return a single and concrete sentence describing the camera moving trajectory by the object's progression in the frames. You don't need to mention all the objects. It is good to describe the moving trajectory without all of the objects.
In-context Examples:
Example 1:
<0>: In the study. there is a plant...to the left of the current spot...
<1>: In the study. there is a bookshelf to the left of the current spot...
<2>: In the hallway. there is a door to the left of the current spot...
Your moving trajectory description: Exit the study. Move from left to right, start near plant, laptop, and table, pass a bookshelf...
Example 2:
...
Your turn:
<0>: In the ***bedroom***, there is a **bed, blanket, table** *to the right of the current spot...*
...
<5>: In the ***bedroom***, here is a **wall** *to the left of the current spot* in near distance...
Your moving trajectory description:

Move through the corridor and step forwards. Pass by a paint on the wall. Then advance through the **bedroom** and into the **bathroom**. Start with a view of the **bed, curtains, and lamp** in the distance. *Move closer*, noting the **headboard, nightstand, and chandelier**. *Pass* the **dresser** ...

(c) Controllable Instruction Generation

Figure 2. Instruction generation in a controllable way. (a) Using open-sourced expert models, we identify *what objects are in the frames*, and assess *how far an object is* and determine *where an object is located*. The information is then textualized to create richly detailed frame captions. (b) BLIP-2 is adopted to predict and smooth room location across sequential frames. (c) Combining room locations and object information, we use GPT-4 for controllable and open-vocabulary instruction generation. The prompt consists of a task instruction that defines the generation task, and in-context examples that constrain the output style.

pose three expert models and design a textual template, *i.e.*, "There is a *object tag* to the *spatial position* of current spot in *relative distance*", to organize the multi-source information to ease GPT generation. Firstly, we used RAM [66] to annotate the object categories within the frames. Based on these category tags, we employed Grounding DINO [42] to locate the objects in the frames. Subsequently, we used Depth-Anything [64] to predict the depth maps corresponding to the frames.

Using this data, we identify the spatial locations and distances of objects relative to the current camera position. By analyzing object bounding box center positions and depth map locations, we can generate frame captions, as illustrated in Figure 2. Finally, objects in different frames can be easily correlated and capture the progression across different frames. More details in the spatial awareness data generation are provided in Appendix C.

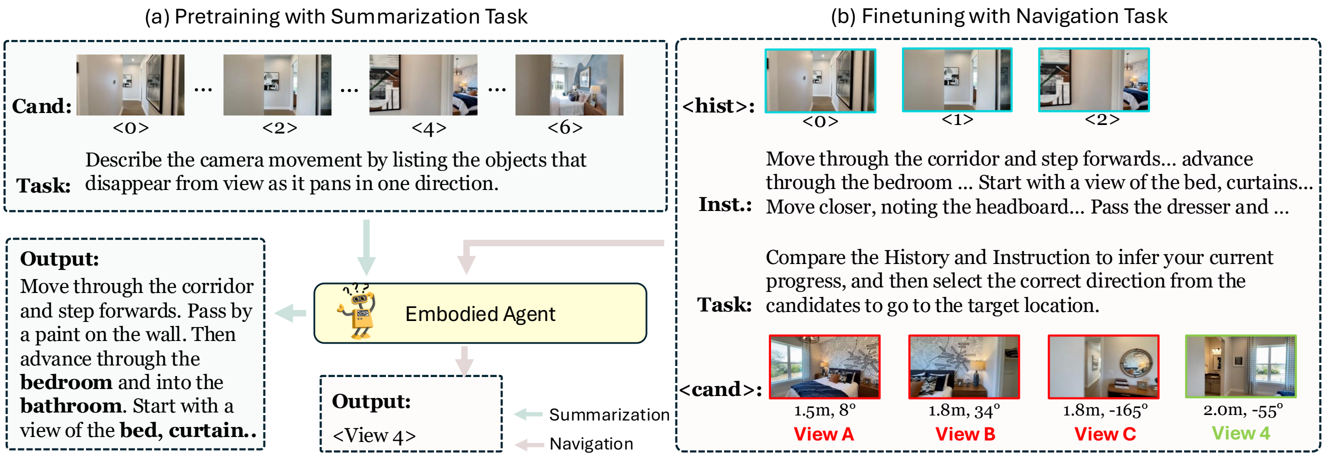
Room Location Annotation in Videos. To determine the camera of each frame, w.r.t. the room category, we used BLIP2 [33] in visual question-answering mode, posing the question, "Which room am I in?" A predefined list of 16 common room types (e.g., bedroom, bathroom, kitchen) was used as possible answers. This list was curated by analyzing 10 randomly sampled long videos, using BLIP2 in generative mode to identify and rank the most frequently mentioned room types. For frame-level predictions, we switched to BLIP2's discriminative mode and applied temporal smoothing to denoise outputs.

We validated this approach by manually annotating 50 video clips, achieving an accuracy of 85%. The use of BLIP2 leverages its open-world knowledge, while limiting room types to 16 categories for discriminative selection simplifies outputs and reduces ambiguity. Any loss in open-vocabulary flexibility is addressed during GPT-based trajectory summarization.

Controllable Instruction Generation. To generate descriptions that accurately capture human-walking trajectories and reflect the environment, we integrate frame-level room locations with frame captions composed of object descriptions. We then employ GPT-4-Turbo [44] for controllable instruction generation, leveraging the multi-source information contained in the composed captions. As depicted in Figure 2, we organize the prompt using the "Task instruction - In-context examples - Prediction" scheme. This approach defines our instruction generation task as describing object progression along the moving trajectories, and includes two examples to ensure GPT produces instruction-style texts only. As shown in Figure 2 (c), captions of frames along the trajectory are embedded into the prompt and input into GPT.

3.2. Action-Enriched Trajectories

3D Environment Reconstruction. To obtain the geometric information of trajectories within RoomTour3D, we employ COLMAP [50, 51] for 3D reconstruction. This process allows us to infer the 3D layout of environments in the videos, providing a detailed geometric context for navigation tasks. Specifically, we sample the videos at 3 frames per second to balance accuracy and execution time. To further improve time efficiency, we split the videos into 100-second video clips with 10-second overlaps between adjacent clips and



Describe the camera movement by listing the objects that disappear from view as it pans in one direction.

Move through the corridor and step forwards. Pass by a paint on the wall. Then advance through the **bedroom** and into the **bathroom**. Start with a view of the **bed, curtain..**

Move through the corridor and step forwards... advance through the bedroom ... Start with a view of the bed, curtains... Move closer, noting the headboard... Pass the dresser and ...

Compare the History and Instruction to infer your current progress, and then select the correct direction from the candidates to go to the target location.

Figure 3. Model training diagram with RoomTour3D. We design two tasks for our RoomTour3D to boost NaviLLM. (a) Pretraining: Sampled frames on the trajectory are treated as candidate observations. Model is optimized to summarize object progression along the path. (b) Finetuning: Each frame is considered as a navigable step. Given historical observation <0> to <2> and navigation instruction, the model is prompted to predict the next action by selecting from candidate observations View A, View B, View C and View 4.

perform reconstruction on the clips simultaneously.

Following this, we merge the resulting sub-models reconstructed from the video clips. For every two adjacent clips, if the reconstructed models have more than three overlapping frames, the models are merged and readjusted into one. However, due to varying reconstruction quality, a single video clip can produce more than one model. To manage this, we construct a graph for merging these sub-models. Each model is treated as a node, and we connect two models if they have more than three overlapping frames. We then apply Depth-First Search [54] to merge any two connected models and replace the original model nodes with the new merged one, continuing this process until no connected nodes remain.

Navigable Action Sampling. We enhance navigation action diversity by using frames from room tour videos as navigable actions by sampling at significant view-change points within a small radius. These points are identified by reconstructing 3D scenes to measure camera orientation differences and distances between frames, capturing varied views from revisited locations or turning points. Frames with substantial view changes are retained using cosine similarity thresholding, followed by non-maximum suppression to isolate major shifts. DBSCAN clustering [11] groups spatially close frames with different views to ensure diverse navigable actions. These measures ensure the robustness to the misidentification of significant view change points that may arise due to inaccuracies in 3D reconstruction. Finally, we identify distinct walking paths within each cluster. For each path, we select the most recent frame as a positive candidate and the frame with the highest angular difference as a negative candidate, enhancing the diversity of navigable actions. More detailed description of our navigable points sampling approach are provided in Appendix B.

4. Vision-and-Language Navigation Model

In this section, we introduce a practice to use our data to train a generalist embodied agent. To start with, we first provide a concise introduction to the state-of-the-art VLN model, NaviLLM [68], which is an LLM-based navigation agent. Then, we introduce two tasks that are adapted for our RoomTour3D data, *i.e.*, vision-instruction summarization task for pretraining and action-instruction navigation task for finetuning.

4.1. Revisiting NaviLLM

NaviLLM is a SOTA LLM-based model for embodied navigation, excelling on benchmarks like CVDN and SOON. It processes panoramic inputs by encoding environmental views and integrating them with navigation instructions. Specific tokens are defined for different types of inputs: <hist> for historical observations and <cand> for candidate views at each navigational step.

During training, NaviLLM receives instructions and a sequence of candidate views. At each step, the candidate observations are input into the model along with the instructions. The model predicts the next action by selecting the appropriate view from the candidates, and this selected view is then cached as a <hist> token, updating the model's internal state for future decisions. At the last step of the navigation task, the model summarizes all <hist> tokens as a separate training task to ensure comprehensive understanding and retention of the navigational history. For testing, the model similarly uses historical observations, *i.e.*, <hist> tokens, accumulated during navigation, and candidate views, *i.e.*, <cand>, at each step to decide the next action. This process ensures that the model's actions align with the given instructions and the observed environment.

4.2. Summarization Task for Pretraining

To leverage the rich, sequential nature of videos and enhance future planning capabilities, we adapt NaviLLM to use the description-enriched trajectories from our RoomTour3D dataset for a summarization task. Each frame is treated as a candidate view and wrapped with <cand> tokens. Similar to selected panoramic views at each navigation step, these frames are considered as selected actions when executing navigation instructions. As shown in Figure 3(a), the frame tokens and summarization task instruction are organized into a unified prompt and input into the LLM. The model is expected to output a trajectory summary containing object progression and room locations, as specified by the task instruction. The model is trained using next-token prediction loss, consistent with the original language model training methodology.

4.3. Navigation Task for Finetuning

In order to enable learning navigation decision-making from scalable scenes, we adapt NaviLLM to action-enriched trajectories from our RoomTour3D. Unlike panoramic views capturing observations from a single location, our data provides candidate views from frames at different locations and orientations, with only one frame directed toward the destination. Each frame in the video sequence is treated as a potential navigable action and wrapped with <cand> tokens. These candidate views are presented to the model and processed in the same way as panoramic views. As shown in Figure 3(b), the model processes the inputs to predict the next action, selecting the appropriate frame from the candidate views. The selected action is then cached as a <hist> token for subsequent decision-making steps.

During fine-tuning, each frame is treated as a navigable step, with the next trajectory frame as the target action and <STOP> as an alternative. The model uses historical observations and navigation instructions to iteratively predict the next action, building a detailed understanding of the path. At the final step, the model summarizes the navigation path, incorporating object progression and room locations. This summarization task enhances its ability to recall navigational history and improves performance.

5. Tasks and Experiments

This section outlines our experimental setup and presents the results. Detailed implementation information can be found in Appendix E.

Datasets. During pretraining, we follow practice from NaviLLM [68] and perform teacher-forcing training on the combined dataset from our video-instruction data from RoomTour3D, together with CVDN [55], SOON [71], R2R [2], REVERIE [46] and ScanQA [3], and augmented data from R2R and REVERIE. In the multi-task fine-tuning stage, we alternate between teacher forcing and student forcing on the combined data from our action-instruction data from RoomTour3D, together with CVDN, SOON, R2R, REVERIE, ScanQA and LLaVA-23k [40].

To evaluate the impact of our data on navigation agent training, we test on CVDN, SOON, R2R, and REVERIE. CVDN requires navigating towards a target by understanding dialog history, linking dialogue comprehension to actions. SOON tasks the agent with locating objects without bounding boxes, emphasizing semantic-visual alignment. R2R involves following step-by-step instructions, requiring dynamic progress tracking and precise alignment with navigational history. REVERIE focuses on localizing distant objects based on concise instructions, aided by ground truth bounding boxes at waypoints.

Evaluation Metrics. For the navigation tasks, we follow the evaluation methodology from [2] using the following navigation metrics: **Success Rate (SR)**, which measures whether the agent reaches the target location within a set distance threshold; **Success Rate Weighted by Path Length (SPL)**, which is the SR adjusted by the ratio of the ground truth path length to the actual path traveled; **Goal Progress (GP)**, the advancement in meters towards the goal. GP is utilized for the CVDN dataset, whereas SR and SPL are the metrics for other datasets.

5.1. Comparison on Supervised Tasks

As shown in Table 1, we performed a one-time fine-tuning on the four tasks in a fully supervised manner. To begin, our experiments reiterate the superiority of multitask training over single-task training. Also, incorporating our RoomTour3D data into the pre-training process led to consistent improvements across all metrics on Val-U, achieving state-of-the-art results in the GP metric in the CVDN dataset.

Notably, finetuning with our action-enriched data results in state-of-the-art performance on both Val-U and Test sets across SOON, R2R and REVERIE tasks. While the improvement on the CVDN and SOON datasets is modest, the most significant boost compared to the reproduced baseline is observed in R2R Val-U and REVERIE Val-U, with gains of approximately 5.7% and 6%, respectively. The improvement in R2R is largely driven by enhanced spatial awareness, stemming from the inclusion of proximity data, which helps the model better understand object distance and position. Similarly, gains in REVERIE are attributed to a combination of open-vocabulary tags, spatial awareness, and the addition of room type data, which encourages the model to infer the layout of environments, thereby boosting its spatial reasoning capabilities. Moreover, our use of open-ended instructions allows the model to adapt flexibly to diverse

Table 1. Overall comparison with the baseline methods. Our RoomTour3D data can boost NaviLLM by a margin on SOON, R2R and REVERIE on SPL metric and on CVDN GP metric. *denotes reproduced results. RT3D$_{Desc}$ and RT3D$_{Action}$ stand for description-enriched trajectories only and action-enriched trajectories.

Methods	**CVDN**		**SOON**		**R2R**		**REVERIE**	
	Val-U	Test	Val-U	Test	Val-U	Test	Val-U	Test
Models Focusing on Single Task								
PREVALENT [20]	3.15	2.44	-	-	53	51	-	-
HOP [47]	4.41	3.24	-	-	57	59	26.1	24.3
HAMT [6]	5.13	5.58	-	-	61	60	30.2	26.7
DUET [7]	-	-	22.6	21.4	60	58	33.7	36.0
VLN-SIG [29]	5.52	5.83	-	-	62	60	-	-
VLN-PETL [48]	5.69	6.13	-	-	60	58	27.7	26.7
NavGPT2 [70]	-	-	-	-	61	60	-	-
BEV-BERT [1]	-	-	-	-	**64**	60	36.4	**36.4**
Unified Model For All Tasks								
NaviLLM(w. Pretrain) [68]	6.16	**7.90**	29.2	26.3	59	60	35.7	32.3
NaviLLM(w. Pretrain)*	6.09	-	28.0	-	56.7	-	31.4	-
NaviLLM+RT3D$_{Desc}$(Ours)	**6.96**	7.55	30.2	26.5	62.3	61.8	37.1	35.1
NaviLLM+RT3D$_{Action}$(Ours)	6.33	7.22	**31.7**	**27.8**	62.4	**62.2**	**37.4**	**36.4**

Table 2. Ablation study on the input modalities for trajectory summarization task.

Object tags	**Depth & Bounding Box**	**Room type**	**CVDN**	**SOON**		**R2R**		**REVERIE**	
			GP↑	SR↑	SPL↑	SR↑	SPL↑	SR↑	SPL↑
×	×	×	6.09	33.64	28.01	65.52	56.67	38.32	31.35
✓	×	×	5.41	32.52	26.51	63.61	55.76	42.52	34.37
✓	✓	×	6.49	37.62	**30.40**	68.37	61.70	41.72	36.04
✓	✓	✓	**6.96**	**38.80**	30.21	**69.37**	**62.28**	**43.25**	**37.10**

Table 3. Overall comparison with SOTA zero-shot methods on R2R. † denotes training exclusive navigable actions. * denotes using 36 views setting. RT3D stands for our RoomTour3D.

Methods	**Val Unseen**	
	SR↑	SPL↑
Random Walk [45]	3	2
Commercial Model		
NavGPT(GPT-3.5) [69]*	13.89	9.12
NavGPT(GPT-4) [69]	34	29
MapGPT(GPT-4) [5]	38.8	25.8
MapGPT(GPT-4V) [5]	**43.7**	34.8
DiscussNav(GPT-4) [43]	43	**40**
Open-source Model		
LangNav(LLaMA2-7B) [45]	0	0
NavCoT(LLaMA2-7B) [37]	7.78	6.50
DuET (Init. LXMERT [52])	1	0
NaviLLM [68]†	0	0
NaviLLM+RT3D(Ours)	**14.33**	**10.86**

scenarios, fostering more robust and generalizable performance and better contextual understanding.

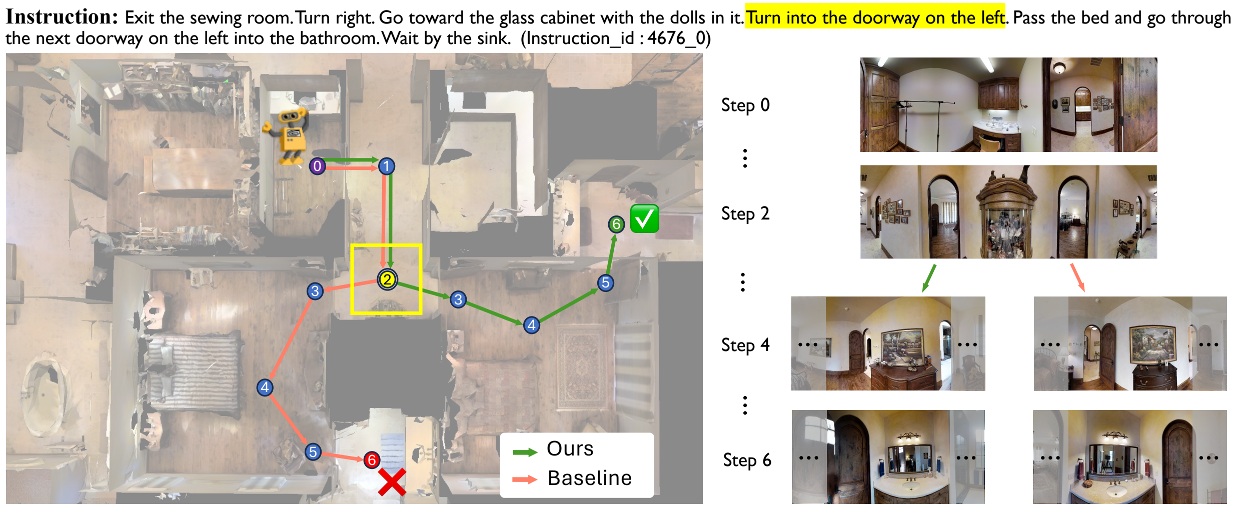
5.2. Comparison on Zero-shot Task

To further demonstrate the substantial indoor knowledge contained in our data and its effectiveness for embodied action and language instructions, we conduct zero-shot experiments on embodied action prediction, as shown in Table 3.

We removed all action and geometric data from the training datasets and retrained NaviLLM with and without our RoomTour3D dataset. Without action prediction data, NaviLLM lacked the ability to learn effective navigable action selection. However, with the inclusion of our action-enriched trajectories, NaviLLM achieved a 14.33% SR and a 10.86% SPL, outperforming open-source models built on LLaMA-7B and reaching results comparable to NavGPT [69], which leverages GPT-3.5. These improvements validate the effectiveness of our 3D trajectories mined from room tour video reconstructions and emphasize the value of our action-enriched trajectories. This highlights the significant contribution of our dataset to advancing open-world navigation.

5.3. Ablation study

Effect of open-world semantics and spatial awareness. As shown in Table 2, we analyzed the impact of various information types on instruction generation. Adding object



Instruction: Exit the sewing room. Turn right. Go toward the glass cabinet with the dolls in it. Turn into the doorway on the left. Pass the bed and go through the next doorway on the left into the bathroom. Wait by the sink. (Instruction_id : 4676_0)

Figure 4. Paths of NaviLLM [68] and ours on R2R-unseen. Purple and green circles denote the start and target locations, respectively, and the red circle represents incorrect endpoint. According to the instruction, the agent should turn left at the waypoint marked with yellow. Our method makes the correct decision, while the baseline is confused by similar entrance at the waypoint, thus mistakenly turns right.

variety significantly improved performance on REVERIE with increase SPL from 31.35% to 34.37%, as this dataset relies on object grounding. However, it had no direct impact on SOON, possibly because SOON relies solely on detailed textual descriptions without explicit bounding box annotations. After introducing depth estimation, which helps determine the relative distances of objects, the performance on SOON, R2R and REVERIE achieve marginal boosts. This demonstrates that enhancing spatial awareness significantly contributes to indoor navigation tasks. Furthermore, incorporating room locations, which capture the scene semantics along the trajectory, provided a moderate boost across all four VLN tasks. This further highlights the critical role of object variety and spatial awareness in improving navigation performance.

Effect of action-instruction data. As shown in Table 1, we test the effect of incorporating video-action-instruction data into the training dataset. It is evident that this approach improves the SPL metric across the test scenes of SOON, R2R, and REVERIE. We believe that incorporating geometric information and movement-inclusive instructions helps the model better align the relationship between action and observation changes, thereby further enhancing the model's embodied capabilities.

5.4. Data correctness verification

We evaluated the correctness of our automated data-generation pipeline by manually rating 100 randomly sampled trajectory descriptions on a 4-point relevance scale: 1 for "totally irrelevant", 2 for "partially relevant", 3 for "mostly relevant" and 4 for a "perfect match". The evaluation yielded an average score of 3.08, with 74% of descriptions rated as "mostly relevant" or "perfect match," demonstrating the method's effectiveness in generating meaningful, visually aligned descriptions.

5.5. Navigation Case Visualization

As shown in Figure 4, selecting the correct action is critical at specific decision points, such as when a left turn is required to follow the instruction accurately. In the example, at step ②, both the rooms to the left and right could satisfy the latter part of the instruction, "pass the bed and go into the bathroom." However, the baseline method incorrectly chooses a right turn at the designated left-turn point, causing it to deviate from the intended path. Once this error occurs, even with scene graph history, the model struggles to realign with the correct trajectory. This challenge is particularly common in household environments, where bedroom layouts often appear similar. It further demonstrates the effectiveness of our data alignment in improving adherence to action-based instructions.

6. Conclusion

In this paper, we present RoomTour3D, a novel dataset automatically curated from room tour videos for VLN tasks. By leveraging the rich, sequential nature of video data and incorporating object variety and spatial awareness, we generate 200K navigation instructions and 17K action-enriched trajectories from 1847 room tour scenes. Additionally, we produce navigable trajectories from video frames and reconstructed 3D scenes, which significantly boost the performance and set new state-of-the-art results on the SOON and REVERIE benchmarks. This approach also enables the development of a trainable zero-shot navigation agent, demonstrating the effectiveness and scalability of RoomTour3D in advancing VLN research.

References

[1] Dong An, Yuankai Qi, Yangguang Li, Yan Huang, Liang Wang, Tieniu Tan, and Jing Shao. BEVbert: Multimodal map pre-training for language-guided navigation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2737–2748, 2023. 2, 7

[2] Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian Reid, Stephen Gould, and Anton van den Hengel. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments, 2018. 1, 2, 6

[3] Daichi Azuma, Taiki Miyanishi, Shuhei Kurita, and Motoaki Kawanabe. ScanQA: 3D question answering for spatial scene understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 6, 17

[4] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from rgb-d data in indoor environments. *International Conference on 3D Vision (3DV)*, 2017. 1, 17

[5] Jiaqi Chen, Bingqian Lin, Ran Xu, Zhenhua Chai, Xiaodan Liang, and Kwan-Yee K Wong. Mapgpt: Map-guided prompting for unified vision-and-language navigation. *arXiv preprint arXiv:2401.07314*, 2024. 3, 7

[6] Shizhe Chen, Pierre-Louis Guhur, Cordelia Schmid, and Ivan Laptev. History aware multimodal transformer for vision-and-language navigation. In *NeurIPS*, 2021. 2, 7

[7] Shizhe Chen, Pierre-Louis Guhur, Makarand Tapaswi, Cordelia Schmid, and Ivan Laptev. Think global, act local: Dual-scale graph transformer for vision-and-language navigation, 2022. 2, 7

[8] Shizhe Chen, Pierre-Louis Guhur, Makarand Tapaswi, Cordelia Schmid, and Ivan Laptev. Learning from unlabeled 3d environments for vision-and-language navigation, 2022. 2

[9] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, 2023. 16

[10] Abhishek Das, Samyak Datta, Georgia Gkioxari, Stefan Lee, Devi Parikh, and Dhruv Batra. Embodied question answering, 2017. 2

[11] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, page 226–231. AAAI Press, 1996. 5, 12

[12] Daniel Fried, Ronghang Hu, Volkan Cirik, Anna Rohrbach, Jacob Andreas, Louis-Philippe Morency, Taylor Berg-Kirkpatrick, Kate Saenko, Dan Klein, and Trevor Darrell. Speaker-follower models for vision-and-language navigation, 2018. 2

[13] Tsu-Jui Fu, Xin Eric Wang, Matthew Peterson, Scott Grafton, Miguel Eckstein, and William Yang Wang. Counterfactual vision-and-language navigation via adversarial path sampling, 2020. 2

[14] Chen Gao, Xingyu Peng, Mi Yan, He Wang, Lirong Yang, Haibing Ren, Hongsheng Li, and Si Liu. Adaptive zone-aware hierarchical planner for vision-language navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14911–14920, 2023. 2

[15] Pierre-Louis Guhur, Makarand Tapaswi, Shizhe Chen, Ivan Laptev, and Cordelia Schmid. Airbert: In-domain pretraining for vision-and-language navigation, 2021. 1, 2

[16] Mingfei Han, Yali Wang, Xiaojun Chang, and Yu Qiao. Mining inter-video proposal relations for video object detection. In *European conference on computer vision*, pages 431–446. Springer, 2020.

[17] Mingfei Han, Yali Wang, Zhihui Li, Lina Yao, Xiaojun Chang, and Yu Qiao. Html: Hybrid temporal-scale multimodal learning framework for referring video object segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. IEEE, 2023.

[18] Mingfei Han, Linjie Yang, Xiaojun Chang, Lina Yao and Heng Wang. Shot2story: A new benchmark for comprehensive understanding of multi-shot videos. *arXiv preprint arXiv:2311.17043*, 2023.

[19] Mingfei Han, Yali Wang, Mingjie Li, Xiaojun Chang, Yi Yang, and Yu Qiao. Progressive frame-proposal mining for weakly supervised video object detection. *IEEE Transactions on Image Processing*, 33:1560–1573, 2024.

[20] Weituo Hao, Chunyuan Li, Xiujun Li, Lawrence Carin, and Jianfeng Gao. Towards learning a generic agent for vision-and-language navigation via pre-training. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13137–13146, 2020. 2, 7

[21] Yicong Hong, Qi Wu, Yuankai Qi, Cristian Rodriguez-Opazo, and Stephen Gould. A recurrent vision-and-language BERT for navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1643–1653, 2021. 2

[22] Yining Hong, Haoyu Zhen, Peihao Chen, Shuhong Zheng, Yilun Du, Zhenfang Chen, and Chuang Gan. 3D-LLM: Injecting the 3d world into large language models, 2023.

[23] Minyoung Hwang, Jaeyeon Jeong, Minsoo Kim, Yoonseon Oh, and Songhwai Oh. Meta-explore: Exploratory hierarchical vision-and-language navigation using scene object spectrum grounding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6683–6693, 2023.

[24] Minyoung Hwang, Jaeyeon Jeong, Minsoo Kim, Yoonseon Oh, and Songhwai Oh. Meta-explore: Exploratory hierarchical vision-and-language navigation using scene object spectrum grounding, 2023. 2

[25] Vihan Jain, Gabriel Magalhaes, Alexander Ku, Ashish Vaswani, Eugene Ie, and Jason Baldridge. Stay on the path: Instruction fidelity in vision-and-language navigation. *arXiv preprint arXiv:1905.12255*, 2019. 1

[26] Aishwarya Kamath, Peter Anderson, Su Wang, Jing Yu Koh, Alexander Ku, Austin Waters, Yinfei Yang, Jason

Baldridge, and Zarana Parekh. A new path: Scaling vision-and-language navigation with synthetic instructions and imitation learning, 2023. 2

[27] Jing Yu Koh, Honglak Lee, Yinfei Yang, Jason Baldridge, and Peter Anderson. Pathdreamer: A world model for indoor navigation, 2021. 2

[28] Alexander Ku, Peter Anderson, Roma Patel, Eugene Ie, and Jason Baldridge. Room-across-room: Multilingual vision-and-language navigation with dense spatiotemporal grounding. *arXiv preprint arXiv:2010.07954*, 2020. 1, 2

[29] Jialu Li and Mohit Bansal. Improving vision-and-language navigation by generating future-view image semantics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10803–10812, 2023. 2, 7

[30] Jialu Li and Mohit Bansal. Panogen: Text-conditioned panoramic environment generation for vision-and-language navigation. *Advances in Neural Information Processing Systems*, 36:21878–21894, 2023. 2

[31] Jialu Li, Hao Tan, and Mohit Bansal. Improving cross-modal alignment in vision language navigation via syntactic information, 2021. 2

[32] Jialu Li, Hao Tan, and Mohit Bansal. EnvEdit: Environment editing for vision-and-language navigation, 2022. 2

[33] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. BLIP-2: Bootstrapping language-image pre-training with frozen image encoders and large language models, 2023. 3, 4

[34] Xiangyang Li, Zihan Wang, Jiahao Yang, Yaowei Wang, and Shuqiang Jiang. KERM: Knowledge enhanced reasoning for vision-and-language navigation, 2023. 2

[35] Xiwen Liang, Fengda Zhu, Lingling Li, Hang Xu, and Xiaodan Liang. Visual-language navigation pretraining via prompt-based environmental self-exploration, 2022. 2

[36] Xiwen Liang, Liang Ma, Shanshan Guo, Jianhua Han, Hang Xu, Shikui Ma, and Xiaodan Liang. CorNav: Autonomous agent with self-corrected planning for zero-shot vision-and-language navigation, 2024. 1

[37] Bingqian Lin, Yunshuang Nie, Ziming Wei, Jiaqi Chen, Shikui Ma, Jianhua Han, Hang Xu, Xiaojun Chang, and Xiaodan Liang. NavCoT: Boosting llm-based vision-and-language navigation via learning disentangled reasoning. *arXiv preprint arXiv:2403.07376*, 2024. 3, 7

[38] Kunyang Lin, Peihao Chen, Diwei Huang, Thomas H. Li, Mingkui Tan, and Chuang Gan. Learning vision-and-language navigation from youtube videos, 2023. 1, 2, 12

[39] Chong Liu, Fengda Zhu, Xiaojun Chang, Xiaodan Liang, Zongyuan Ge, and Yi-Dong Shen. Vision-language navigation with random environmental mixup, 2021. 2

[40] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning, 2023. 6

[41] Rui Liu, Wenguan Wang, and Yi Yang. Volumetric environment representation for vision-language navigation. In *CVPR*, pages 16317–16328, 2024. 2

[42] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, et al. Grounding Dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499*, 2023. 2, 3, 4, 13

[43] Yuxing Long, Xiaoqi Li, Wenzhe Cai, and Hao Dong. Discuss before moving: Visual language navigation via multi-expert discussions. *arXiv preprint arXiv:2309.11382*, 2023. 2, 3, 7

[44] OpenAI. GPT-4: Generative pre-trained transformer 4. OpenAI API, 2022. 2, 3, 4, 12

[45] Bowen Pan, Rameswar Panda, SouYoung Jin, Rogerio Feris, Aude Oliva, Phillip Isola, and Yoon Kim. LangNav: Language as a perceptual representation for navigation. *arXiv preprint arXiv:2310.07889*, 2023. 3, 7

[46] Yuankai Qi, Qi Wu, Peter Anderson, Xin Wang, William Yang Wang, Chunhua Shen, and Anton van den Hengel. REVERIE: Remote embodied visual referring expression in real indoor environments, 2020. 1, 2, 6

[47] Yanyuan Qiao, Yuankai Qi, Yicong Hong, Zheng Yu, Peng Wang, and Qi Wu. HOP: History-and-order aware pre-training for vision-and-language navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15418–15427, 2022. 2, 7

[48] Yanyuan Qiao, Zheng Yu, and Qi Wu. VLN-PETL: Parameter-efficient transfer learning for vision-and-language navigation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15443–15452, 2023. 2, 7

[49] Santhosh K Ramakrishnan, Aaron Gokaslan, Erik Wijmans, Oleksandr Maksymets, Alex Clegg, John Turner, Eric Undersander, Wojciech Galuba, Andrew Westbury, Angel X Chang, et al. Habitat-Matterport 3D dataset (HM3D): 1000 large-scale 3d environments for embodied ai. *arXiv preprint arXiv:2109.08238*, 2021. 1

[50] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-Motion Revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 1, 4, 13, 14

[51] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision (ECCV)*, 2016. 1, 4, 13, 14

[52] Hao Tan and Mohit Bansal. Lxmert: Learning cross-modality encoder representations from transformers. *arXiv preprint arXiv:1908.07490*, 2019. 7

[53] Hao Tan, Licheng Yu, and Mohit Bansal. Learning to navigate unseen environments: Back translation with environmental dropout, 2019. 2

[54] Robert Tarjan. Depth-first search and linear graph algorithms. In *12th Annual Symposium on Switching and Automata Theory (swat 1971)*, pages 114–121, 1971. 5

[55] Jesse Thomason, Michael Murray, Maya Cakmak, and Luke Zettlemoyer. Vision-and-dialog navigation, 2019. 1, 2, 6

[56] Liuyi Wang, Zongtao He, Ronghao Dang, Mengjiao Shen, Chengju Liu, and Qijun Chen. Vision-and-language navigation via causal learning. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 2

[57] Su Wang, Ceslee Montgomery, Jordi Orbay, Vighnesh Birodkar, Aleksandra Faust, Izzeddin Gur, Natasha Jaques, Austin

Waters, Jason Baldridge, and Peter Anderson. Less is more: Generating grounded navigation instructions from landmarks, 2022. 2

[58] Xin Eric Wang, Vihan Jain, Eugene Ie, William Yang Wang, Zornitsa Kozareva, and Sujith Ravi. Environment-agnostic multitask learning for natural language grounded navigation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIV 16*, pages 413–430. Springer, 2020. 1

[59] Zun Wang, Jialu Li, Yicong Hong, Yi Wang, Qi Wu, Mohit Bansal, Stephen Gould, Hao Tan, and Yu Qiao. Scaling data generation in vision-and-language navigation. In *ICCV 2023*, 2023. 1, 2

[60] Yuetian Weng, Mingfei Han, Haoyu He, Xiaojun Chang, and Bohan Zhuang. Longvlm: Efficient long video understanding via large language models. *arXiv preprint arXiv:2404.03384*, 2024.

[61] Erik Wijmans, Samyak Datta, Oleksandr Maksymets, Abhishek Das, Georgia Gkioxari, Stefan Lee, Irfan Essa, Devi Parikh, and Dhruv Batra. Embodied question answering in photorealistic environments with point cloud perception, 2019. 2

[62] Wikipedia. Preferred walking speed. https://en.wikipedia.org/wiki/Preferred_walking_speed. 3

[63] Fei Xia, Amir R Zamir, Zhiyang He, Alexander Sax, Jitendra Malik, and Silvio Savarese. Gibson Env: Real-world perception for embodied agents. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9068–9079, 2018. 1

[64] Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth Anything: Unleashing the power of large-scale unlabeled data. In *CVPR*, 2024. 2, 3, 4, 13

[65] Jiazhao Zhang, Kunyu Wang, Rongtao Xu, Gengze Zhou, Yicong Hong, Xiaomeng Fang, Qi Wu, Zhizheng Zhang, and Wang He. NaVid: Video-based vlm plans the next step for vision-and-language navigation. *arXiv preprint arXiv:2402.15852*, 2024. 1

[66] Youcai Zhang, Xinyu Huang, Jinyu Ma, Zhaoyang Li, Zhaochuan Luo, Yanchun Xie, Yuzhuo Qin, Tong Luo, Yaqian Li, Shilong Liu, et al. Recognize Anything: A strong image tagging model. *arXiv preprint arXiv:2306.03514*, 2023. 2, 3, 4, 12, 13

[67] Ganlong Zhao, Guanbin Li, Weikai Chen, and Yizhou Yu. Over-nav: Elevating iterative vision-and-language navigation with open-vocabulary detection and structured representation, 2024. 2

[68] Duo Zheng, Shijia Huang, Lin Zhao, Yiwu Zhong, and Liwei Wang. Towards learning a generalist model for embodied navigation, 2023. 2, 5, 6, 7, 8, 16

[69] Gengze Zhou, Yicong Hong, and Qi Wu. NavGPT: Explicit reasoning in vision-and-language navigation with large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 7641–7649, 2024. 3, 7

[70] Gengze Zhou, Yicong Hong, Zun Wang, Xin Eric Wang, and Qi Wu. Navgpt-2: Unleashing navigational reasoning capability for large vision-language models. In *European Conference on Computer Vision*, pages 260–278. Springer, 2025. 7

[71] Fengda Zhu, Xiwen Liang, Yi Zhu, Xiaojun Chang, and Xiaodan Liang. SOON: Scenario oriented object navigation with graph-based exploration, 2021. 1, 2, 6

Appendix

The indexes of figures and tables in the appendix are continuous to the main sections for easy reference.

Dataset release. Our annotations and intermediate products are released at https://huggingface.co/datasets/roomtour3d/roomtour3d under CC-BY-SA-4.0 license. The downscaled and sampled video frames are released at https://huggingface.co/datasets/roomtour3d/room_tour_video_3fps under CC-BY-SA-4.0 license. The codes and project updates are hosted at https://roomtour3d.github.io/.

Overview. In the supplementary material, we provide

- **Section A:** Room tour video collection process.
- **Section B:** Navigable point extraction used for action-enriched trajectory generation.
- **Section C:** Object variety and spatial awareness for trajectory descriptions.
- **Section D:** Room tour 3D scene reconstruction.
- **Section E:** Further model implementation details.
- **Section F:** Qualitative results showcasing the instruction following capabilities of our trained model.
- **Section G:** Data samples and excerpts from our data verification report to illustrate data curation correctness.
- **Section H:** Broader impact of our work, including limitations and future extendable works.

A. Room Tour Video Collection

To enable more diversity for indoor scenes, we leveraged the rich variety and volume of room tour videos available on YouTube. These videos, recorded with hand-held cameras from a first-person perspective, offer a realistic and dynamic view of indoor environments. We curated a dataset from 1847 YouTube room tour videos, in total 243 hours. Our data collection approach builds on the video list from YTB-VLN [38], which we further filtered and expanded to enhance diversity and quality.

To ensure high-quality data, we prioritize continuous videos with least transitions, such as human interviews or abrupt cutting into close-ups, for better 3D reconstruction. We applied a title-description-based filtering process by using GPT-4 [44] and excluded videos shorter than three minutes. Additionally, we detected abrupt video transitions, retaining videos with at least nine continuous shots occupying over 80% of the video duration. We further extended our dataset by continuously updating high-quality channels (*e.g.*, NavaRealtyGroup, Open House 24, Sona Visual) with new videos, resulting in our current 1847 room tour scenes.

To process this data, we spatially downscale the resolution to shorter side 360 and temporally downsample the frame rate to 3 frames per second. All the following processing are performed on this downsampled data.

B. Navigable points generation

To inject open-world knowledge from room tour videos into navigation agents, we propose navigating agents using video frames. Each frame in a human walking demonstration can be treated as having two next actions: move forward or stop. However, at significant view-change points — instances of distinct view shifts within a close radius — we sample frames with varied orientations as candidate actions to enhance the agent's training. Unlike YTB-VLN [38], which composes panoramic images at room nodes, our approach involves taking every significant view-change point and its neighboring frames that meet specific criteria as candidate actions.

First, we detect significant view-change points along person's trajectory. By reconstructing the 3D scene, we can determine the camera orientation difference and distance between frames. There are instances where the person may revisit a nearly identical location, resulting in varied views within almost the same spatial region. Additionally, turning points with notable view changes in close proximity are essential to capture. Identifying these view-change points is useful for producing diversified navigable action data, especially when panorama images are not available.

To find these points, for each point along the trajectory we calculated pairwise cosine similarity. We then applied a threshold of 45 degrees to retain only frames that demonstrate a substantial change in view. Afterwards, non-maximum suppression is performed along the trajectory to isolate local maxima in angular change to highlight the most significant view changes.

To account for the points that are close in proximity, but have different views due to an intersection in the walking trajectory, we performed DBSCAN clustering [11] of the points that were retained after Non-Maximum Suppression. This clustering step ensures a diverse set of navigable actions is maintained, even without the availability of panoramic images.

Finally, as shown in Figure 5, to extract varied navigable action candidates, we post-processed the clusters by identifying the distinct walking paths of the person within each cluster. In cases where paths intersect, the cluster may encompass two separate routes. For each walking path, we select the most recent frame on the walking path as a positive candidate, while a negative candidate is chosen as the frame within the cluster that exhibits the highest angular difference in view with the positive candidate.

C. Instruction Generation

In this section, we detail the process of transforming spatial awareness and object variety information into textual captions for use with GPT. This involves extracting multi-source data using models such as RAM (Swin-L) [66],

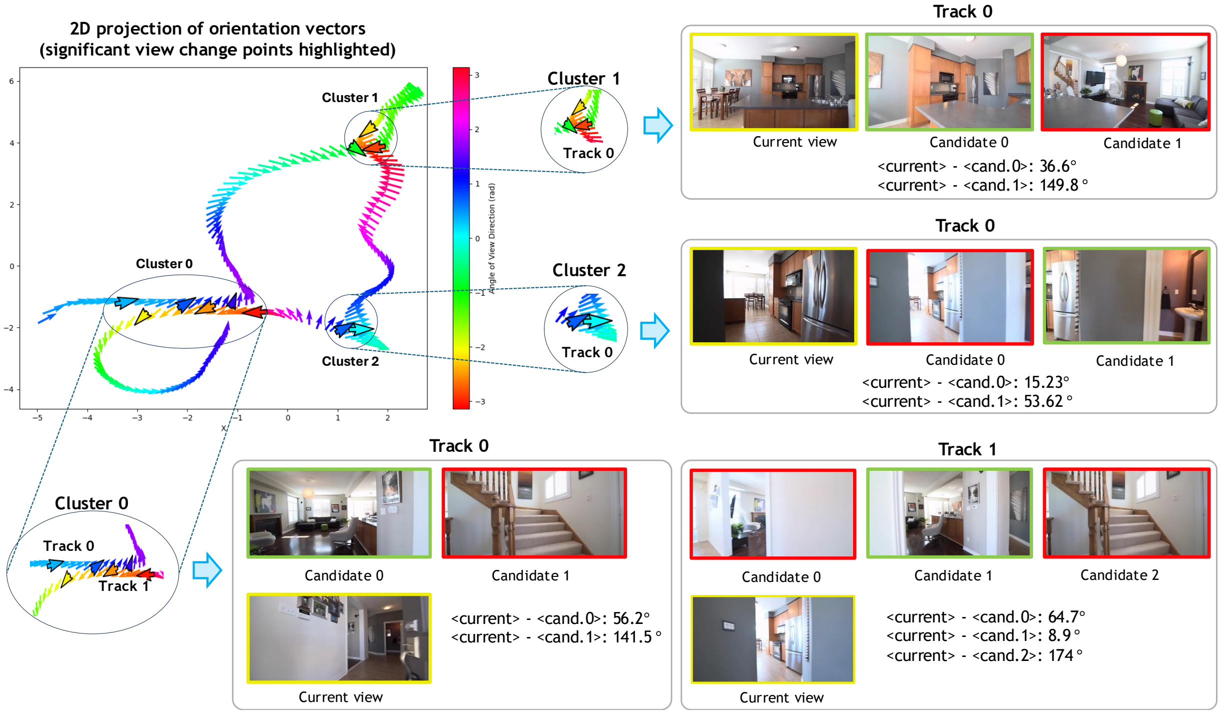


Figure 5. Visualization of significant view change point selection. For each cluster we identify the walking tracks and find the candidate views for the next action selection. This process ensures we have a diversified set of views in the setting without panorama images.

Grounding DINO [42], and Depth-Anything [64], and then organizing this information into structured text inputs.

Object variety into texts. Web videos offer a rich, open-world setup, capturing diverse items, arrangements, room functionalities, and layouts, which are critical for training open-world navigation agents. To fully utilize this diversity and ensure a controllable generation of instructions, we use RAM [66] (Swin-L) to extract object tags in each frame. For each frame, we filter out the resulting entries indicating room types in order to be consistent with the identified room locations from BLIP-2. Then these object tags are used for grounding objects within the frames, for further integration of spatial awareness information.

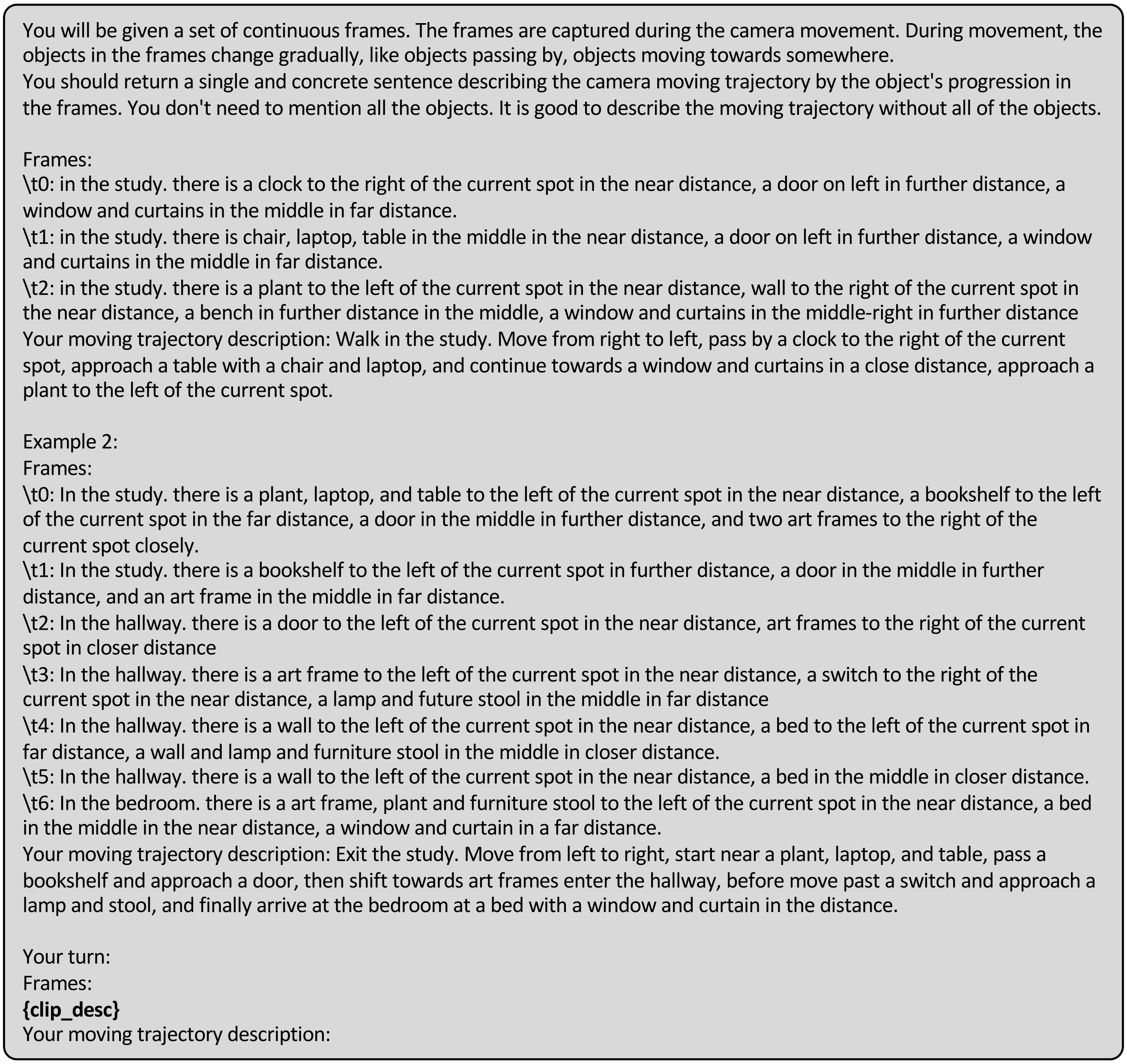
Spatial awareness into texts. Navigation agents are frequently tasked with approaching or obtaining objects, making it crucial for them to sense object locations and dynamics during movement. To achieve this, we jointly use Grounding DINO [42] and Depth-Anything [64] models to gather detailed spatial information. The reason why we used Depth-Anything over the depth derived from COLMAP [50, 51] reconstructions is its ability to directly extract reliable depth without relying on long-range frames or structure-from-motion processes, which are prone to errors in complex video reconstructions. This spatial awareness information is then transformed into text inputs suitable for GPT, enabling effective training.

We start by using Grounding DINO to spatially localize objects within the video frames. We define spatial locations relative to the capturing spot: *to the left of the current spot*, *in the middle*, and *to the right of the current spot*. Specifically, the center 40% of the frame is considered the middle, the leftmost 30% as the left, and the rightmost 30% as the right. For depth perception, we categorize distances into three ranges: *in the near distance* (closest 30%), *in closer distance* (next 40%), and *in a further distance* (remaining 30%).

Followingly, we integrate spatial location and depth estimation by measuring the overlap ratio between objects and the defined distance ranges. For example, if a carpet overlaps with the near-distance area by more than 30%, we consider the carpet to be in the near distance to the capturing spot. Large objects that span multiple distance categories, such as a carpet visible in both near, closer, and further distances, are annotated accordingly to reflect their extended presence within the scene.

This structured approach ensures that our instructions capture the relative positioning and depth of objects, providing comprehensive context for navigation tasks. These texts are then further organized into GPT to generate contextually rich instructions for navigation agents training.

GPT generation. We utilize GPT-4 to summarize the object progression during the walking trajectory, leveraging the detailed object variety and spatial awareness texts. The template used for organizing the components is depicted in



You will be given a set of continuous frames. The frames are captured during the camera movement. During movement, the objects in the frames change gradually, like objects passing by, objects moving towards somewhere.

You should return a single and concrete sentence describing the camera moving trajectory by the object's progression in the frames. You don't need to mention all the objects. It is good to describe the moving trajectory without all of the objects.

Frames:

\t0: in the study. there is a clock to the right of the current spot in the near distance, a door on left in further distance, a window and curtains in the middle in far distance.

\t1: in the study. there is chair, laptop, table in the middle in the near distance, a door on left in further distance, a window and curtains in the middle in far distance.

\t2: in the study. there is a plant to the left of the current spot in the near distance, wall to the right of the current spot in the near distance, a bench in further distance in the middle, a window and curtains in the middle-right in further distance

Your moving trajectory description: Walk in the study. Move from right to left, pass by a clock to the right of the current spot, approach a table with a chair and laptop, and continue towards a window and curtains in a close distance, approach a plant to the left of the current spot.

Example 2:

Frames:

\t0: In the study. there is a plant, laptop, and table to the left of the current spot in the near distance, a bookshelf to the left of the current spot in the far distance, a door in the middle in further distance, and two art frames to the right of the current spot closely.

\t1: In the study. there is a bookshelf to the left of the current spot in further distance, a door in the middle in further distance, and an art frame in the middle in far distance.

\t2: In the hallway. there is a door to the left of the current spot in the near distance, art frames to the right of the current spot in closer distance

\t3: In the hallway. there is a art frame to the left of the current spot in the near distance, a switch to the right of the current spot in the near distance, a lamp and future stool in the middle in far distance

\t4: In the hallway. there is a wall to the left of the current spot in the near distance, a bed to the left of the current spot in far distance, a wall and lamp and furniture stool in the middle in closer distance.

\t5: In the hallway. there is a wall to the left of the current spot in the near distance, a bed in the middle in closer distance.

\t6: In the bedroom. there is a art frame, plant and furniture stool to the left of the current spot in the near distance, a bed in the middle in the near distance, a window and curtain in a far distance.

Your moving trajectory description: Exit the study. Move from left to right, start near a plant, laptop, and table, pass a bookshelf and approach a door, then shift towards art frames enter the hallway, before move past a switch and approach a lamp and stool, and finally arrive at the bedroom at a bed with a window and curtain in the distance.

Your turn:

Frames:

{clip_desc}

Your moving trajectory description:

Figure 6. Prompt used for GPT-based instruction generation. We provide instruction for this generation task, in-context examples.

Figure 6. For each clip, we organize the object tags, spatial locations, relative distance to the camera and room locations per frame. This arranged content is then fed into GPT-4 to generate the trajectory summary and instructions. For data sample visualization, please refer to Sec. G.

D. Room Reconstruction

To obtain complete geometric information, we adopt COLMAP [50, 51] for indoor reconstruction. In this subsection, we detail the procedure of reconstructing room tour scenes, which further facilitates sampling navigable frames.

Reconstruction of video clip. To reconstruct video clips, we start by sampling videos at 3 frames per second to balance accuracy and execution time. This frame rate provides sufficient detail for accurate 3D reconstruction while maintaining manageable processing times. Each video is divided into 100-second clips with a 10-second overlap between adjacent clips. Using COLMAP, we perform structure-from-motion and multi-view stereo processing on each clip. It estimates camera poses and generates a sparse 3D point cloud by identifying and matching feature points across frames. The command used for reconstruction is shown as follows, in which ‘$DATASET_PATH’ denotes the folder containing

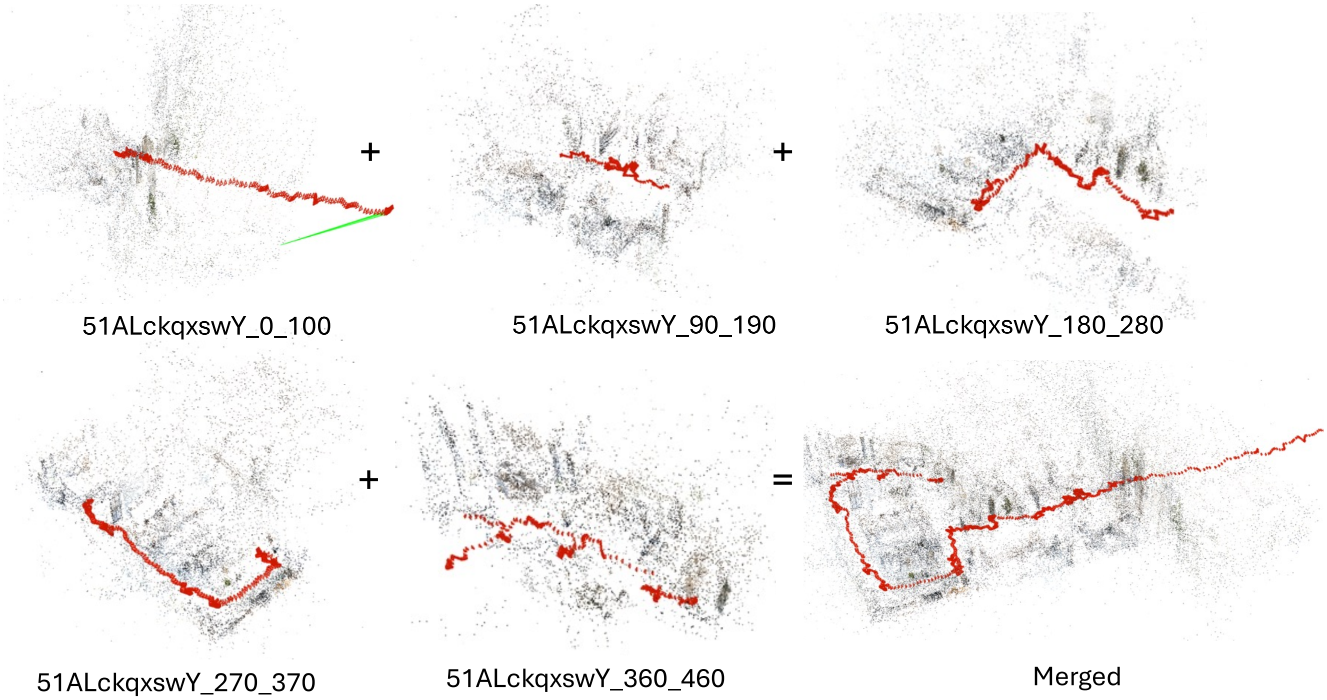


Figure 7. Illustration of the COLMAP model merging process. Reconstructed models from 5 adjacent video clips are successfully merged into one holistic model.

sub-clip frames and reconstructed models will be located.

```
colmap automatic_reconstructor \
    --image_path $DATASET_PATH/$IMG_FOLDER\
    --workspace_path $DATASET_PATH \
    --data_type individual \
    --quality high \
    --single_camera 1 \
    --sparse 1 \
    --dense 0 \
    --num_threads 10 --use_gpu 0
```

COLMAP model merging After performing individual reconstructions on video clips, we proceed to merge the resulting COLMAP models to create a unified 3D representation of the room tour scenes, as shown in Figure 7.

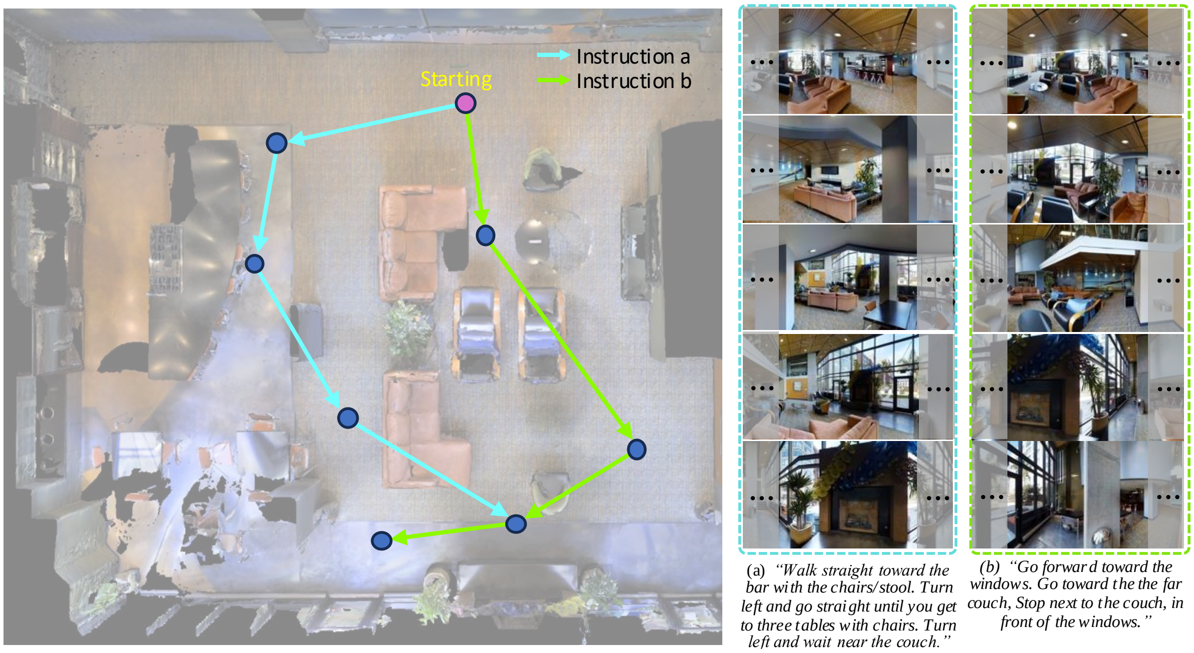
We begin by identifying overlapping frames between adjacent clips. These overlapping frames serve as common reference points for aligning and merging the separate models. If two reconstructed models share more than 3 common frames, we will try to merge these two models using the command as the following, where model merging and bundle adjustment are conducted in sequence.

```
colmap model_merger \
    --input_path1 $MODEL_1 \
    --input_path2 $MODEL_2 \
    --output_path $RESULTED_MODEL_BEF_BA

colmap bundle_adjuster \
    --input_path $RESULTED_MODEL_BEF_BA \
    --output_path $RESULTED_MODEL_AFT_BA
```

However, due to potential variances in reconstruction quality, a single video clip may produce multiple sub-models. To handle this, we adopt a graph-based approach for merging, *i.e.*, Depth-First Search. In this approach, each submodel is represented as a node in the graph. Edges are created between nodes that share more than three overlapping frames, indicating that these sub-models can be merged.

We iteratively merge the model nodes with edge connection existing by traversing from the first video clip (*e.g.*, clip "0_100"). The successfully merged model will be a new graph node to replace the original separated two nodes. In order to monitor the quality of this model merging operation, we use reprojection error to determine whether rolling back the merging operation. Specifically, if the error of the merged model is even larger than the sum of the two separate models, the model merging operation will be rolled back. This iterative merging process continues until no further connections exist, resulting in a comprehensive and continuous 3D model of the room tour scenes. The final merged model provides detailed geometric information that is crucial for accurately sampling navigable frames and enhancing the training data for navigation agents.



(a) *"Walk straight toward the bar with the chairs/stool. Turn left and go straight until you get to three tables with chairs. Turn left and wait near the couch."*

(b) "Go forward toward the windows. Go toward the the far couch, Stop next to the couch, in front of the windows."

Figure 8. Visualization of the method trained with RoomTour3D on unseen scene *8194nk5LbLH* with trajectory ID 4332. The agent successfully follows navigation instructions in R2R dataset. In (a), the agent first moves towards the bar and then approaches the couch. In (b), the agent moves forward towards the windows, then proceeds to the far sofa, and finally stops in front of the window.

E. Implementation details

Following the practice from NaviLLM [68], we fine-tune the multi-view fusion module and the LLM. The multi-view fusion module consists of a 2-layer transformer encoder with a hidden size of 1024, and the LLM is built upon Vicuna-7B-v1.1 [9]. The ViT in the scene encoder is EVA-CLIP-02-Large, which remains frozen during training. Our training follows a two-stage strategy using the Adam optimizer with a learning rate 3e-5. The model is trained for 2500 steps in the pre-training stage and 1250 steps in the multi-task fine-tuning stage, with a batch size 256. The training process utilizes 4×8 Nvidia A100 GPUs. During testing, we employ a sampling strategy with a temperature of 0.01 for the SOON and REVERIE tasks to encourage exploration, while a greedy strategy is used for other tasks. This approach ensures robust performance across various evaluation scenarios.

F. Qualitative Results

This section presents qualitative results to demonstrate the effectiveness of our model trained with the RoomTour3D dataset. The model was evaluated on unseen scenes using the R2R dataset, focusing on its performance in following navigation instructions. As shown in Figure 8, we tested the model on an unseen scene, *8194nk5LbLH*, with trajectory ID 4332. Experimented with two different instructions, the agent trained our data shows its flexibility in following the instructions. For example, in (a), the agent moves straight to the bar, then reaches the three tables with chars, and finally stops near the couch. In (b), the agent directly moves towards the window, following the instructions, then moves towards the far coach and stops. These results demonstrate the instruction-following navigation ability of the agent, which further highlights the effectiveness of our video-instruction data.

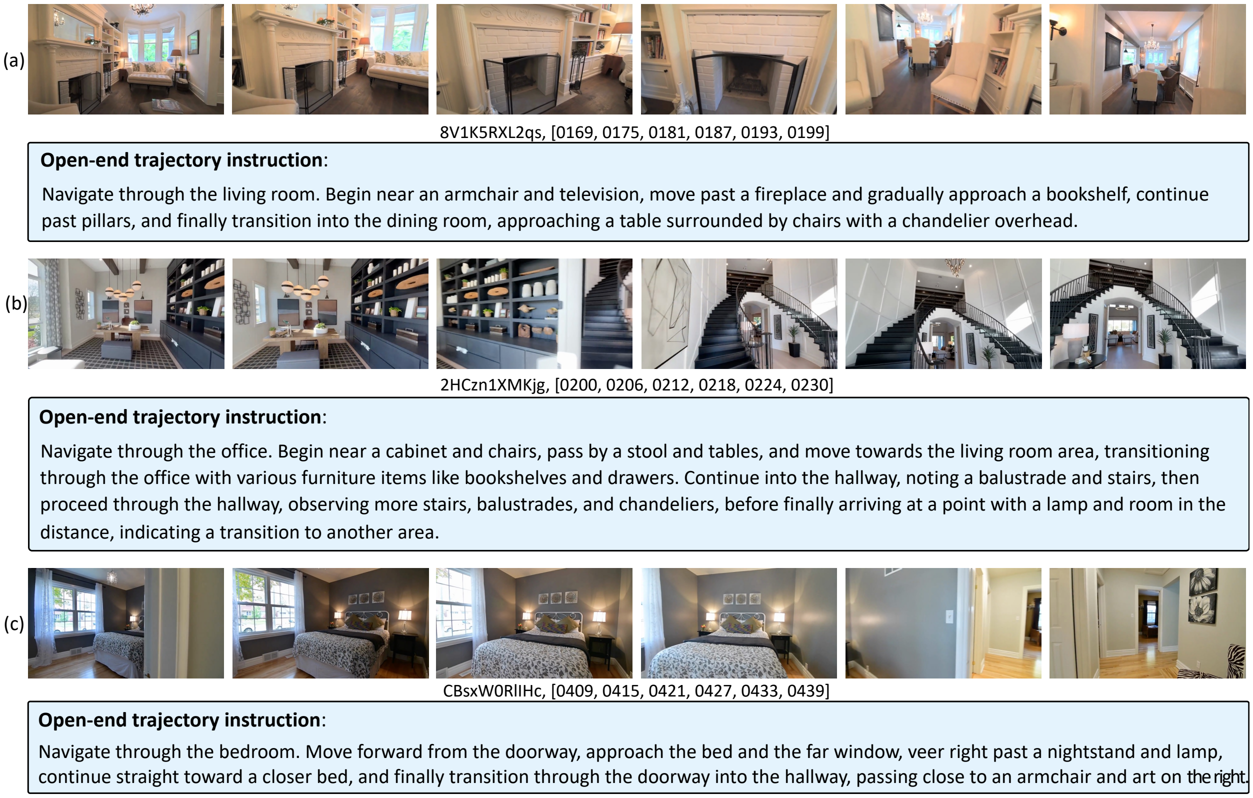
G. Data Sample Visualization

In this section, we present visualizations of data samples from the RoomTour3D dataset, as shown in Figure 9. These visualizations highlight the rich variety of indoor scenes, the spatial awareness embedded in the data, and the detailed annotations used for training navigation agents.

Data correctness verification. We provide part (14 out of 100) of manual check trajectories in Figure 10 and Figure 11. For each trajectory, sampled frames and generated descriptions are provided, along with the manual check scores. The score ranges from 1 to 4, representing "totally irrelevant", "partially relevant", "mostly relevant" and "perfect match" respectively. Most of the sampled trajectories gain scores 3 and 4, which shows the convincing quality of our automatically generated descriptions.

H. Broader Impact

Data Limitations and Ethical Considerations. We provide downsampled video frames instead of the original videos. Users can also download these from the original sources. Additionally, our meticulous filtering process en-


(a)
8V1K5RXL2qs, [0169, 0175, 0181, 0187, 0193, 0199]
(b)
2HCzn1XMKjg, [0200, 0206, 0212, 0218, 0224, 0230]
(c)
CBsxW0RlIHc, [0409, 0415, 0421, 0427, 0433, 0439]

Open-end trajectory instruction:

Navigate through the living room. Begin near an armchair and television, move past a fireplace and gradually approach a bookshelf, continue past pillars, and finally transition into the dining room, approaching a table surrounded by chairs with a chandelier overhead.

Open-end trajectory instruction:

Navigate through the office. Begin near a cabinet and chairs, pass by a stool and tables, and move towards the living room area, transitioning through the office with various furniture items like bookshelves and drawers. Continue into the hallway, noting a balustrade and stairs, then proceed through the hallway, observing more stairs, balustrades, and chandeliers, before finally arriving at a point with a lamp and room in the distance, indicating a transition to another area.

Open-end trajectory instruction:

Navigate through the bedroom. Move forward from the doorway, approach the bed and the far window, veer right past a nightstand and lamp, continue straight toward a closer bed, and finally transition through the doorway into the hallway, passing close to an armchair and art on the right.

Figure 9. Example open-ended trajectories and instructions. The instruction captures the surrounding environments and the object dynamics ("move past a fireplace" in (a), "move towards the living room area" in (b)), and more importantly, the moving directions and destination ("approaching a table surrounded by chairs" in (a), "into the hallway, passing close to" in (c)). All these data are automatically generated without manual correction.

sures that the video frames and annotations contain only indoor rooms and houses, containing no personally identifiable information or offensive content. The authors will take responsibility for long-term maintenance.

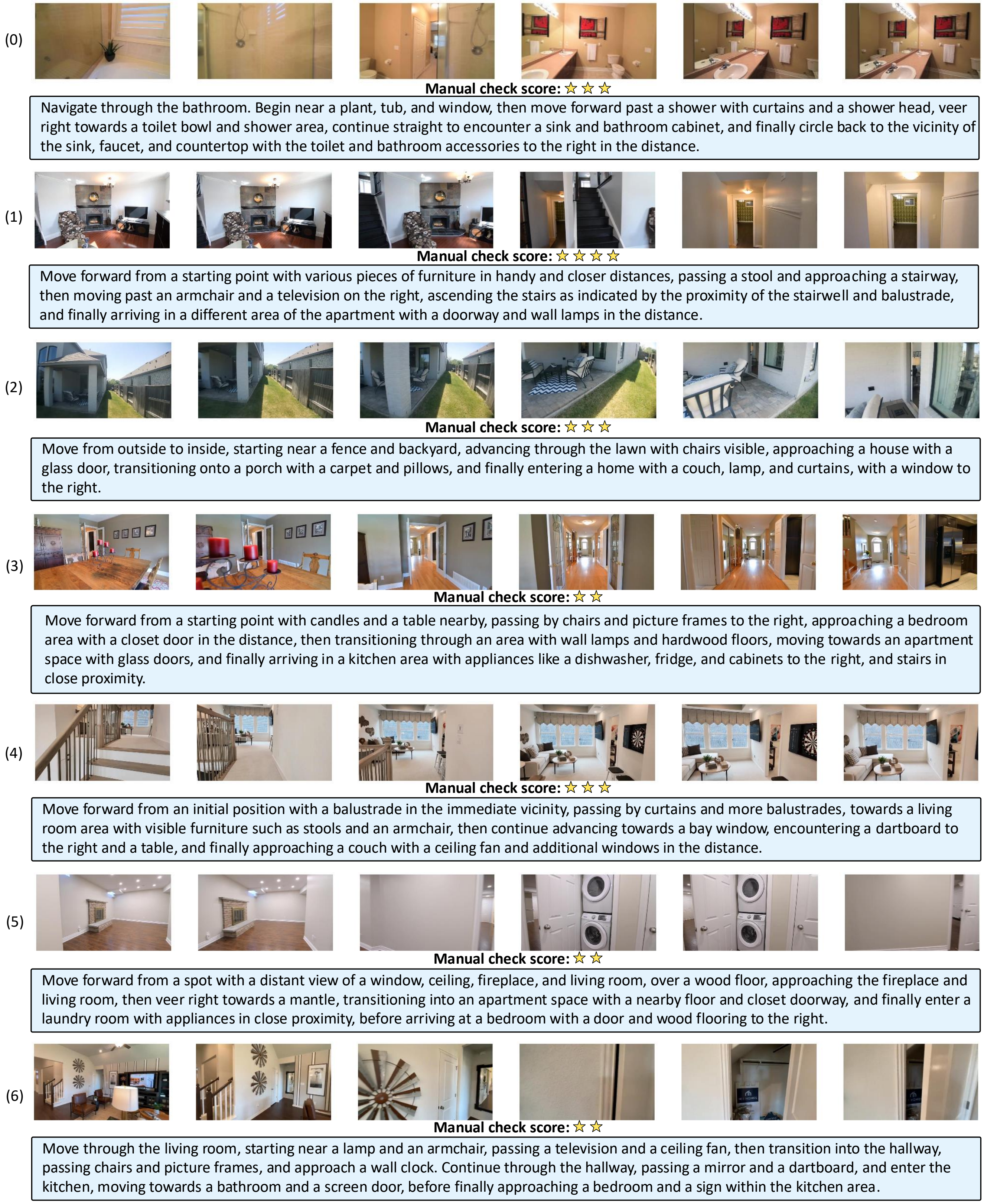
Scope of Conclusions. It is important to recognize that experiments and data, including ours, might only represent a subset of universal realities. Nevertheless, given the wide range of room tour scenes covered in our videos, we believe our conclusions offer a robust understanding applicable to indoor embodied navigation. While specific to our dataset and results, these findings provide significant insight into the broader field of embodied navigation.

Usage of Language Models and Simulators. Our use of the LLaMA model[1] from Meta, use of MatterPort3D data [4] is authorized for research purposes. Those intending to use our model post-release should ensure they have the necessary permissions and adhere to usage restrictions. We express deep respect for the work of developers and contributors, recognizing their integral role in advancing language modeling and data collection.

Future Research and Development. Aligned with our commitment to the research community, we released our code and dataset. This is intended to encourage further research and enable others to build upon our work. Although our current experiments require up to 8×4 A100-80G GPUs for pretraining and 8 A100-80G for multi-task tuning, we are aware this may be a limitation. Consequently, we plan to focus future efforts on adapting these experiments to be compatible with parameter-efficient tuned LLMs. It's important to note that fitting the experiments within an 8 GPU or fewer framework is not the primary focus of this paper. Still, we consider it a crucial step towards making our research more accessible and inclusive for various research groups.

Also, it would be interesting to investigate the usefulness of our data for grounded question-answering for 3D environments, particularly on the ScanQA dataset [3].

[1] https://llama.meta.com/



(0)

Manual check score: ☆☆☆

Navigate through the bathroom. Begin near a plant, tub, and window, then move forward past a shower with curtains and a shower head, veer right towards a toilet bowl and shower area, continue straight to encounter a sink and bathroom cabinet, and finally circle back to the vicinity of the sink, faucet, and countertop with the toilet and bathroom accessories to the right in the distance.

(1)

Manual check score: ☆☆☆☆

Move forward from a starting point with various pieces of furniture in handy and closer distances, passing a stool and approaching a stairway, then moving past an armchair and a television on the right, ascending the stairs as indicated by the proximity of the stairwell and balustrade, and finally arriving in a different area of the apartment with a doorway and wall lamps in the distance.

(2)

Manual check score: ☆☆☆

Move from outside to inside, starting near a fence and backyard, advancing through the lawn with chairs visible, approaching a house with a glass door, transitioning onto a porch with a carpet and pillows, and finally entering a home with a couch, lamp, and curtains, with a window to the right.

(3)

Manual check score: ☆☆

Move forward from a starting point with candles and a table nearby, passing by chairs and picture frames to the right, approaching a bedroom area with a closet door in the distance, then transitioning through an area with wall lamps and hardwood floors, moving towards an apartment space with glass doors, and finally arriving in a kitchen area with appliances like a dishwasher, fridge, and cabinets to the right, and stairs in close proximity.

(4)

Manual check score: ☆☆☆

Move forward from an initial position with a balustrade in the immediate vicinity, passing by curtains and more balustrades, towards a living room area with visible furniture such as stools and an armchair, then continue advancing towards a bay window, encountering a dartboard to the right and a table, and finally approaching a couch with a ceiling fan and additional windows in the distance.

(5)

Manual check score: ☆☆

Move forward from a spot with a distant view of a window, ceiling, fireplace, and living room, over a wood floor, approaching the fireplace and living room, then veer right towards a mantle, transitioning into an apartment space with a nearby floor and closet doorway, and finally enter a laundry room with appliances in close proximity, before arriving at a bedroom with a door and wood flooring to the right.

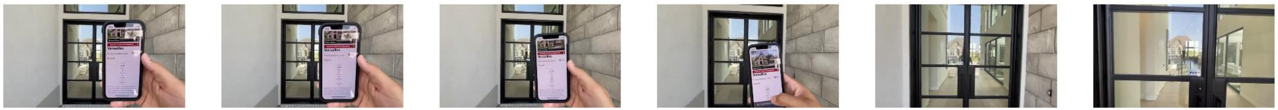
(6)

Manual check score: ☆☆

Move through the living room, starting near a lamp and an armchair, passing a television and a ceiling fan, then transition into the hallway, passing chairs and picture frames, and approach a wall clock. Continue through the hallway, passing a mirror and a dartboard, and enter the kitchen, moving towards a bathroom and a screen door, before finally approaching a bedroom and a sign within the kitchen area.

Figure 10. Trajectory samples for manual check. For each trajectory, we provide frames and descriptions for check. The rating ranges from 1 to 4, representing "totally irrelevant", "partially relevant", "mostly relevant" and "perfect match" respectively. 7 out of 100 samples are shown here.

(7)



Manual check score: ☆ ☆

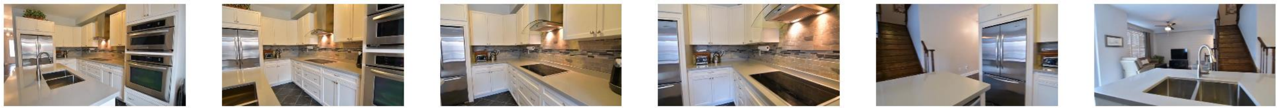
Navigate through the bathroom. Begin near a plant, tub, and window, then move forward past a shower with curtains and a shower head, veer right towards a toilet bowl and shower area, continue straight to encounter a sink and bathroom cabinet, and finally circle back to the vicinity of the sink, faucet, and countertop with the toilet and bathroom accessories to the right in the distance.

(8)

Manual check score: ☆ ☆ ☆ ☆

Navigate through the kitchen into the living room. Begin near the kitchen sink and cabinets, move past appliances and countertops, approach the fridge, and continue past more cabinetry. Transition from the kitchen to the living room, passing a table and wall lamps, and finally arrive in the living room, moving towards a glass door with curtains, a fireplace, and armchairs, with a screen door to the right.

(9)



Manual check score: ☆ ☆ ☆ ☆

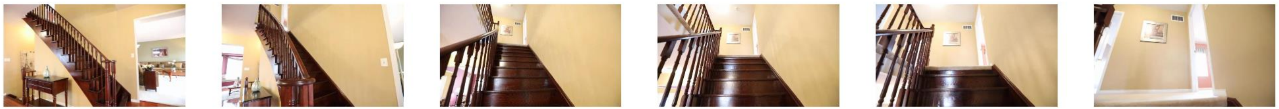
Navigate through the kitchen. Begin near the counter top and microwave, move past various appliances like a dishwasher and exhaust hood, veer right passing closer to the microwave and oven, continue towards the coffee machine and tile wall, then shift towards the sink and exhaust hood on the right, and finally approach the kitchen island with a stool, ending near the kitchen sink with a ceiling fan and stairwell in the distance.

(10)

Manual check score: ☆ ☆

Move forward from a position near a girl and a phone, approaching a bathroom with a mirror and multiple doorways, then pass by a man and various bathroom fixtures such as a faucet, sink, vanity, and bathroom cabinet, before moving through the bathroom door and past curtains and a lamp, and finally turning right towards stairs and stools, indicating a transition from the bathroom area to another room or a stairway.

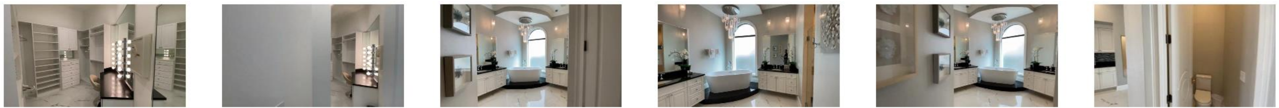
(11)



Manual check score: ☆ ☆ ☆

Navigate through the hallway. Progress forward, initially close to a balustrade, then approach a stairwell, continue past picture frames and rails, and finally head towards a room with a window and doorway in the distance, with the stairwell nearby.

(12)



Manual check score: ☆ ☆ ☆ ☆

Move forward from a bedroom setting, passing by a chair and dresser, towards a bathroom area, gradually approaching a stool and vanity on the right, and finally arriving at a bathroom with a tub, sink, and toilet bowl, with a closet doorway in close proximity.

(13)

Manual check score: ☆ ☆ ☆ ☆

Move from the outside towards a house, starting near basketball hoops, then passing by chairs and a garage door, approaching a driveway and yard, and continuing towards the house exterior and porch. Progress closer to the house, passing more chairs and approaching the doorway and stairs, before finally nearing the entrance with wall lamps, a carpet, and a pillow, indicating arrival at the home's threshold.

Figure 11. Trajectory samples for manual check - Continued. For each trajectory, we provide frames and descriptions for check. The rating ranges from 1 to 4, representing "totally irrelevant", "partially relevant", "mostly relevant" and "perfect match" respectively. 7 out of 100 samples are shown here.