Uni-NaVid: A Video-based Vision-Language-Action Model for Unifying Embodied Navigation Tasks

Jiazhao Zhang[1,2]   Kunyu Wang[3]   Shaoan Wang[1,2]   Minghan Li[2]   Haoran Liu[1,2]
Songlin Wei[1,2]   Zhongyuan Wang[3]   Zhizheng Zhang[2,3,†]   He Wang[1,2,3,†]

[1]CFCS, School of Computer Science, Peking University   [2]Galbot   [3]Beijing Academy of Artificial Intelligence

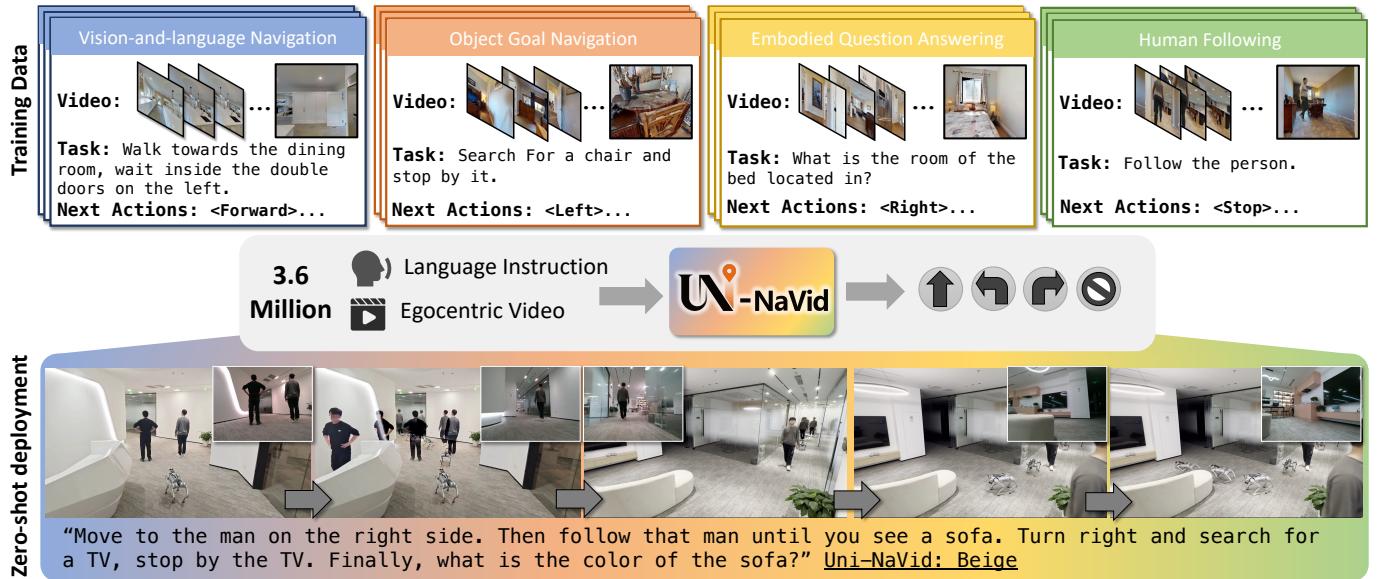https://pku-epic.github.io/Uni-NaVid

Fig. 1: Uni-NaVid learns general navigation skills across four embodied navigation tasks using 3.6 million navigation samples. Uni-NaVid only takes online RGB video frames and language instructions as input and output actions, achieving general navigation ability in a real-world deployment.

Abstract—**Embodied Navigation is a fundamental capability for intelligent robots, requiring robots to follow human commands and move autonomously within physical environments. Despite significant advancements, most existing navigation approaches are tailored to specific navigation tasks, such as instruction following, searching objects, answering questions, tracking people, and more. However, the increasing demands on advanced embodied navigation pose the challenge of designing a practical navigation agent that can incorporate multiple navigation tasks naturally and benefits from the synergy between these tasks. To this end, we present Uni-NaVid, a video-based vision-language-action (VLA) model to unify different paradigms of navigation tasks and improve navigation performance by encouraging the synergy among different navigation sub-tasks. This VLA model can directly take natural language instructions and RGB video streams as inputs and output low-level robotic actions in an end-to-end manner. To efficiently process extensive RGB video streams, we propose an online token merge strategy that spatially and temporally consolidates similar visual information which improves the inference speed to 5 Hz. For training Uni-NaVid, we collect 3.6 million navigation data samples across different navigation tasks. Extensive experiments on diverse navigation benchmarks demonstrate that Uni-NaVid achieves state-of-the-art performance within a unified framework by using only ego-centric RGB video as inputs. Additionally, real-world experiments confirm the model's effectiveness and efficiency, shedding light on its strong generalizability.**

I. Introduction

Embodied navigation [112, 85] is a critical capability for intelligent robots and has drawn significant attention in the robotics community. For successful embodied navigation, robots must be able to move autonomously within physical environments based on human instructions. However, navigation tasks vary significantly, and most existing studies are designed for specific tasks, e.g., vision-and-language navigation [44, 46], object goal navigation [12], embodied question answering [21, 90], and following [113, 36, 68]. Consequently, most current approaches are developed to address only one type of navigation task, often relying on specialized modules and task-specific datasets. This narrow scope limits their applicability to multi-purpose navigation applications and prevents these methods from leveraging potential synergies across diverse navigation tasks.

Developing a versatile navigation model presents significant challenges, as it requires the unification of navigation task

modeling and the integration of heterogeneous data for joint use. Initial efforts adopt imitation learning (IL) [85, 93, 66] or reinforcement learning (RL) [106, 97] to learn general navigation skills in simulation environments or limited diverse real-world environments. However, due to the limited rendering quality and diversity of simulators, these approaches often encounter the "sim-to-real" gap and suffer from poor generalization across diverse navigation tasks [27, 5, 40]. Recent studies [120, 114, 61, 60, 78] have attempted to achieve a higher degree of unification using pre-trained large language models (LLMs). However, due to the low frequency of LLM inference, they simplify the problem to some extent by adopting discretized modeling approaches. They rely on pre-defined graphs for decision-making learning, which sacrifices output flexibility and introduces additional challenges for real-world deployment.

In this work, we propose Uni-NaVid, a video-based Vision-Language-Action (VLA) model for unifying diverse commonly demanded navigation tasks (Tab. I). Uni-NaVid takes egocentric RGB video streams and natural language instructions as inputs, and directly generates low-level actions for navigation in continuous environments. To achieve multi-task navigation while supporting efficient navigation, Uni-NaVid extend video-based VLM [51] by incoparting two key components: (1) an efficient VLA architecture based on an online token merge mechanism, which enables efficient processing of online-captured video streams for LLM inference; and (2) an extensive collection of 3.6M samples across four widely studied navigation tasks. We provide a detailed elaboration below:

During navigation, the agent is required to process a substantial volume of online captured frames, which results in memory overload and computational latency, particularly in LLM-based approaches [111, 61]. To this end, we propose an online token merging mechanism to compress near historical frames with a relatively low ratio while compressing far historical frames with a relatively high ratio. This merging mechanism operates in an on-the-fly manner, maximizing the reuse of previous navigation history. In this way, Uni-NaVid learn compact representations that maintain not only fine-grained spatial information but also structured temporal information, thus speeding up the model inference by reducing the token number. Besides, Uni-NaVid adopts a foresight prediction to generate actions for a future horizon at once instead of step-by-step. This enables Uni-NaVid to achieve 5Hz inference, facilitating the deployment of a non-blocking navigation robot powered by a VLA model in real-world environments (Please refer to the supplementary video).

We aim to build Uni-NaVid as a versatile multi-task navigation agent, incorporating four widely demanded navigation tasks: vision-and-language navigation, object-goal navigation, embodied question answering, and human following. These tasks are distinct from each other, with varying task settings and objectives. Specifically, for the human-following task, we construct a new language-guided human-following benchmark for data collection and evaluation. Finally, we collect 3.6M

Methods	Action		Embodied Navigation Tasks			
	D.E.	C.E.	VLN [44]	ObjNav [76]	EQA [90]	Follow [68]
VLMaps [34]	✓		✓	✓		
NaviLLM [114]	✓		✓	✓	✓	
InstructNav [61]	✓		✓	✓		
Poliformer [106]		✓		✓		✓
Uni-NaVid		✓	✓	✓	✓	✓

TABLE I: **Task and setting comparison.** Uni-NaVid is developed to address four embodied navigation tasks, generating action outputs in continuous environments. C.E.: Continuous Environment; D.E.: Discrete Environment.

navigation samples based on diverse navigation tasks with different simulation environments. Additionally, inspired by the success of manipulation VLAs [9], we further integrate 2.3M real-world internet data samples for Video Question Answering (VQA) [7, 51] and video captioning [19] as auxiliary tasks. This integration aims to enhance scene understanding and promote sim-to-real generalization.

We conduct extensive experiments on benchmarks across the aforementioned four navigation tasks and compared our method with strong baselines specifically designed for each task. Utilizing only RGB video streams and instructions as inputs, our method demonstrates the superiority of a single VLA model across diverse benchmarks, achieving SOTA or SOTA-comparable performance. Furthermore, comprehensive ablation studies validate the synergistic benefits of learning multiple navigation tasks jointly. Finally, real-world experiments demonstrate that Uni-NaVid achieves non-blocking navigation exhibiting impressive robustness in handling diverse instructions and environments. We believe our work serves merely as a starting point for general-purpose navigation, and *we will release the full source code to benefit the community.*

II. RELATED WORKS

Multi-Task Embodied Navigation. Embodied navigation [2, 94, 112] requires agents to navigate in unseen environments based on human instructions. There is extensive literature on embodied navigation; here, we focus on four mainstream tasks that involve both visual information and language instructions: Vision-and-Language Navigation [4, 44, 46], Object Goal Navigation [12], Embodied Question Answering [21], and Human Following [37, 68, 118, 119]. Early efforts [85, 93, 66, 97] towards a generalist-embodied navigation model involved multi-task navigation datasets and directly learning navigation skills, showing initial success in multi-task performance. However, these methods experienced performance drops when deployed in novel environments, especially in real-world settings. In recent years, advanced approaches [114, 61, 35, 60, 121, 78] have leveraged the generalization capabilities of large language models to improve multi-task navigation. These models show promising generalizability across navigation tasks but rely on extensive prompting, which impacts time efficiency. In contrast, our video-based large language model is trained end-to-end for multi-task navigation, offering robust generalization and computational efficiency for tasks like human following.
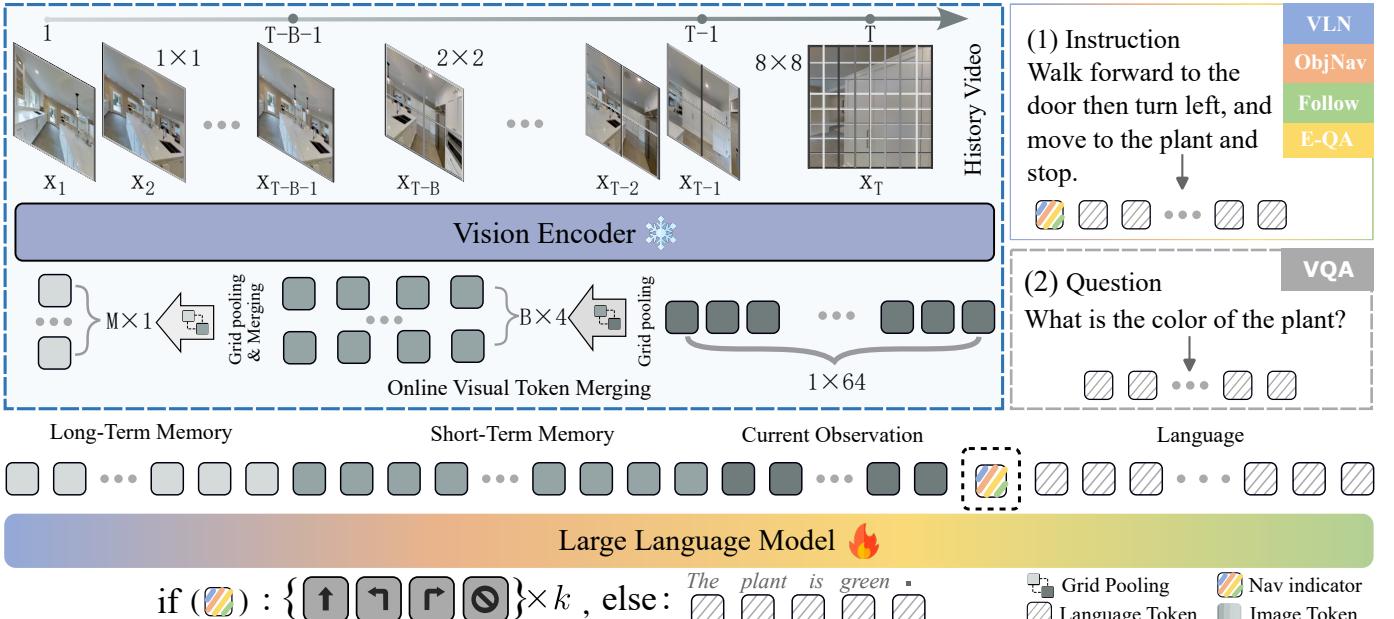
Fig. 2: **Pipeline of Uni-NaVid.** Our method takes only single-view RGB frames $\{\mathbf{x}_1, \cdots, \mathbf{x}_T\}$ and a natural language instruction $\mathcal{I}$ as input. For each frame, we extract 64 visual tokens using the vision encoder and then use online token merging to accelerate the model while retaining compact visual information. The merged tokens and instruction tokens are sent to the large language model to obtain actions for navigation or answers for embodied question-answering.

Embodied Navigation Datasets. To train and evaluate the performance of a policy for embodied navigation tasks, a large body of datasets and corresponding benchmarks have been proposed [23, 123, 59, 64]. These datasets play a crucial role in the embodied navigation community. Here, we review the datasets most relevant to our methods. For vision-and-language navigation, the most widely used datasets are Room-2-Room (R2R) [4] and Room-cross-Room (RxR) [47], which provide navigation instructions and ground truth trajectories of landmarks. We focus on a variant of R2R and RxR in continuous environments, called VLN-CE [44], which is more practical for real-world applications. For object goal navigation, there are several famous benchmarks such as HM3D [69], MP3D [11], and Aithor [124], which are built on various scene environments and simulators. Here, we leverages the HM3D dataset on Habitat [76], which shares the same action settings as VLN-CE. For embodied question answering (EQA), there are diverse datasets focusing on different attributes of EQA, such as MP3D-EQA [90], MT-EQA [103], Graph-EQA [83], and MX-EQA [38]. We select MP3D-EQA, which is well-maintained with the latest baselines. For human-following [116, 117] benchmarks, there is currently no benchmark that provides textual descriptions of humans. Therefore, we have self-built a textual description-based human-following benchmark using Habitat 3.0 [68]. Note that new benchmarks are consistently being proposed, covering a diverse range of navigation attributes. However, our goal is to train and evaluate our method on mainstream datasets to clearly justify the performance of our approach.

Large Language Models for Navigation. Large Language Models (LLMs)[20, 54, 122] have been introduced into robotic navigation due to their generalization capabilities in understanding and planning. One straightforward approach[120, 61, 60, 78] is to use off-the-shelf large language models in a zero-shot manner. These methods employ visual foundation models [22, 54] to describe surrounding environments in text format, prompting the language model to select landmarks that guide the agent. However, abstracting dense visual information into text and relying on discrete landmarks results in sparse environmental observations and is limited to static environments. Another approach [111, 106] trains a video-based large language model end-to-end with low-level actions to enable continuous movement. However, it faces efficiency challenges in long-horizon tasks. In contrast, Uni-NaVid implements an online visual token merging strategy, optimizing training efficiency for long-horizon tasks and supporting non-blocking execution in real-world environments.

III. PROBLEM FORMULATION

Navigation task definition. We define the general-purpose navigation of Uni-NaVid as follows: At the time T, given a natural language instruction $\mathcal{I}$ consisting of l words and an ego-centric RGB video $\mathcal{O}_T$ comprising a sequence of frames $\{\mathbf{x}_1, \cdots, \mathbf{x}_T\}$, the agent is required to plan the next k actions $\{\mathcal{A}_T, \cdots, \mathcal{A}_{T+k-1}\}$ to executed for complete the instruction within novel environments ($k = 4$ in our experiments). Here, we adopt a widely used action setting [76, 12, 44, 21], which require the agent to take low-level actions $\mathbf{a} \in \mathcal{A}$, including

{FORWARD,TURN-LEFT,TURN-RIGHT,STOP}. Note that, our task formulation is compatible with existing embodied navigation tasks [76, 12, 44, 21], where the discrete low-level actions [76, 12, 44, 21] represent a small rotation (30 degrees) or a forward movement (25 cm), making them flexible to be used in continuous environments such obstacle avoidance. We provide a detailed explanation of how these actions are applied in both synthetic and real-world environments in Sec. VI-A.

Overview. As illustrated in Figure 2, Uni-NaVid is composed of three main components: a vision encoder, an online token merge mechanism and a large language model (LLM). First, the online captured video stream is encoded by the vision encoder (EVA-CLIP [82] in implementation) to extract frame-wise visual features in the form of tokens, which we denote them as visual tokens. The visual tokens are then spatially and temporally merged by leveraging an online token merge mechanism. Next, the merged visual tokens are projected with an MLP projector into a feature space aligned with language tokens, which are referred to as visual observation tokens. As common, the instructions are also tokenized as a set of tokens, known as language observation tokens. Both the visual observation tokens and language observation tokens are concatenated and passed to the Large Language Model (LLM), which infers four action tokens that represent the next four actions.

IV. MODEL OF UNI-NAVID

A. Observation Encoding.

Given the ego-centric video up to time T, denoted by $\mathcal{O}_T = \{\mathbf{x}_1 \cdots, \mathbf{x}_T\}$, we encode the video to a sequence of visual features in the form of tokens. For each frame $\mathbf{x}_t$, we first get its visual feature tokens $\mathbf{X}_t \in \mathbb{R}^{N_x \times C}$ with a vision encoder (EVA-CLIP [82] in implementation), where N_x is the patch number (N_x is set to 256) and C is the embedding dimension.

$$\mathbf{X}_{1:T} = Encoder(\mathbf{x}_{1:T}) \qquad (1)$$

The visual features provide rich information that enables the agent to understand its navigation history and plan subsequent actions. However, during navigation, the progressively increasing number of visual tokens ($T \times N_x$) results in progressively longer inference times for the LLM (typically 1–2 seconds per inference) [111]. This increased latency renders LLM-based navigation impractical for deployment in real-world environments.

B. Online Visual Token Merging

To reduce the number of visual tokens while preserving sufficient navigation visual information, we design an token merging mechanism. This strategy is based on the key insight that recent observations are more critical for navigation, and that visual information between consecutive frames (temporally) and within neighboring pixels (spatially) may be redundant.

Visual token grouping. Drawing inspiration from the Atkinson-Shiffrin memory model [6, 80], we categorize visual tokens into current visual tokens $\mathbf{X}_{curr}$, short-term visual tokens $\mathbf{X}_{short}$, and long-term visual tokens $\mathbf{X}_{long}$. These visual tokens are grouped based on their timestamps relative to the current frame T and for each group of visual tokens, we apply a grid pooling operation at different pooling resolutions:

$$\mathbf{X}_{1:T} = \begin{cases} \mathbf{X}_{curr} = GridPool(\mathbf{X}_t, \alpha_{curr}), & \text{if } t = T \\ \mathbf{X}_{short} = GridPool(\mathbf{X}_t, \alpha_{short}), & \text{if } t \in [\text{T-B, T}) \\ \mathbf{X}_{long} = GridPool(\mathbf{X}_t, \alpha_{long}), & \text{if } t \in [1, \text{T-B}) \end{cases}$$
$$(2)$$

where GridPool($\cdot$) is a grid pooling operation [51, 111], *spatially* squeezing the tokens from N_x to $\frac{N_x}{\alpha^2}$, and B (set to 64) is the length of the buffer of shorter memory. Here, we adopt the $\alpha_{curr} = 2$, $\alpha_{short} = 8$, $\alpha_{long} = 16$, leads to visual tokens as $\mathbf{X}_{curr} \in \mathbb{R}^{64 \times C}$, $\mathbf{X}_{short} \in \mathbb{R}^{4 \times C}$, $\mathbf{X}_{long} \in \mathbb{R}^{1 \times C}$, respectively. Here, current visual tokens $\mathbf{X}_{curr}$ encapsulate comprehensive visual information, enabling the agent to perceive its immediate environment and plan subsequent trajectories. Meanwhile, $\mathbf{X}_{short}$ and $\mathbf{X}_{long}$ capture temporally rich information from the captured video stream, facilitating the agent's comprehension of its navigation history.

It should be noted that these hyperparameters are obtained through empirical experimentation to achieve an optimal balance between manageable token numbers and adequate visual information representation. These hyperparameters can be further adjusted when memory capacity and computational resources are not limiting factors. We provide a detailed explanation and ablation study of α in the supplemental material.

Online visual token process. During the navigation process, the agent consistently observes new frames. However, performing encoding and grouping (Eq. 2) for all frames at each step would be computationally intensive. To address this, we implement an online visual token processing mechanism that maximizes the reuse of previously generated visual tokens. Specifically, when a new frame at time $T + 1$ is received, we apply grid pooling exclusively to the most recent visual tokens at time T and the oldest short-term visual tokens at time $T - B$. These processed tokens are then integrated into the short-term and long-term visual tokens, respectively:

$$\mathbf{X}_{curr \to short} = GridPool(\mathbf{X}_{curr}, \frac{\alpha_{short}}{\alpha_{curr}}), \qquad (3)$$

$$\mathbf{X}_{short \to long} = GridPool(\mathbf{X}_{short}, \frac{\alpha_{long}}{\alpha_{short}}). \qquad (4)$$

To prevent the linear growth of long-term visual tokens $\mathbf{X}_{Long}$, we further perform token merging on the long-term visual tokens by combining adjacent tokens that exhibit high similarity, following the approach of VLM-based methods [8, 80]. Specifically, we merge the long-term visual tokens based on the cosine similarity between $\mathbf{X}_{short \to long}$ and the most recent long-term visual tokens $\mathbf{X}_{long}$ at time $T - B - 1$. If the similarity exceeds a predefined threshold τ, we merge them according to the number of frames previously merged (denoted as K) in the latest long-term visual tokens:

Algorithm 1 Online Visual Token Merging

Require:
- Total number of frames T
- Short memory buffer length B
- Grid pooling scales: α_{curr}, α_{short}, α_{long}
- Current visual tokens: $\mathbf{X}_T \in \mathbb{R}^{N_x \times C}$
- Previously merged tokens: $\mathbf{X}_{\text{curr}}$, $\mathbf{X}_{\text{short}}$, $\mathbf{X}_{\text{long}}$
- Number of frames merged in the last tokens of long memory: K

Ensure:
- Updated merged tokens: $\mathbf{X}'_{\text{curr}}$, $\mathbf{X}'_{\text{short}}$, $\mathbf{X}'_{\text{long}}$
- Updated number of frames merged in the last tokens of long memory: K'

1: **if** $T == 1$ **then** ▷ First frame, empty history tokens
2:     $\mathbf{X}'_{\text{short}}, \mathbf{X}'_{\text{long}} \leftarrow []$
3: **else** ▷ Update short-term visual tokens
4:     $\mathbf{X}_{\text{curr}\rightarrow\text{short}} \leftarrow GridPool(\mathbf{X}_{\text{curr}}, \frac{\alpha_{\text{short}}}{\alpha_{\text{curr}}})$
5:     $\mathbf{X}'_{\text{short}} \leftarrow \mathbf{X}_{\text{short}} + [\mathbf{X}_{\text{curr}\rightarrow\text{short}}]$
6: **end if**
7: $\mathbf{X}'_{\text{curr}} \leftarrow GridPool(\mathbf{X}_T, \alpha_{\text{curr}})$ ▷ New current visual token
8: **if** $T > B + 1$ **then** ▷ Out of short-term tokens buffer
9:     $\mathbf{X}_{\text{short}\rightarrow\text{long}} \leftarrow GridPool(\mathbf{X}_{\text{short}}[0], \frac{\alpha_{\text{long}}}{\alpha_{\text{short}}})$
10:     $\mathbf{X}'_{\text{short}} \leftarrow \mathbf{X}_{\text{short}}[1:]$
11:     $s \leftarrow \cos(\mathbf{X}_{\text{long}}[-1], \mathbf{X}_{\text{short}\rightarrow\text{long}})$
12:     **if** $T > B + 2$ and $s > \tau$ **then** ▷ Fuse long-term tokens
13:         $\mathbf{X}_{\text{last\_long}} \leftarrow \frac{1}{K+1}(K\mathbf{X}_{\text{long}}[-1] + \mathbf{X}_{\text{short}\rightarrow\text{long}})$
14:         $\mathbf{X}'_{\text{long}} \leftarrow \mathbf{X}_{\text{long}}[:-1] + [\mathbf{X}_{\text{last\_long}}]$
15:         $K' \leftarrow K + 1$
16:     **else** ▷ Add new long-term token
17:         $\mathbf{X}'_{\text{long}} \leftarrow \mathbf{X}_{\text{long}} + [\mathbf{X}_{\text{short}\rightarrow\text{long}}]$
18:         $K' \leftarrow 1$
19:     **end if**
20: **end if**

$$\mathbf{X}_{\text{long}} = \frac{1}{K+1}\left(K\mathbf{X}_{\text{long}} + \mathbf{X}_{\text{short}\rightarrow\text{long}}\right), \quad (5)$$

$$subject\ to \quad \cos\left(\mathbf{X}_{\text{long}}, \mathbf{X}_{\text{short}\rightarrow\text{long}}\right) > \tau. \quad (6)$$

We insert new long-term visual tokens $\mathbf{X}_{\text{short}\rightarrow\text{long}}$ when their similarity falls below a threshold τ (empirically set to $\tau = 0.95$ [80]), indicating that they contain relatively distinct visual information. This online visual token processing preserves the navigation visual history in a highly compact form (with a length of $M \ll T - B - 1$). Notably, only visual tokens at the boundaries of groups require parallelizable grid pooling, making the process computationally efficient and naturally suited for online deployment in real-world navigation tasks. We give a description of our token merging technique at Algorithmn 1.

Compared to existing video-based large language models [111, 80, 51], this online merging strategy significantly reduces inference time, achieving an average of 0.2 seconds per inference. This improvement becomes increasingly notable when handling longer video sequences. A detailed analysis of time efficiency is provided in the Supplementary Materials.

C. Action Planning

After obtaining the merged visual tokens from semantic features [82], we adopt established practices in Vision-and-Language models [54, 51] to perform vision-language alignment, enabling the large language model (LLM) to effectively interpret visual information. Specifically, we leverage a cross-modality projector $P_V(\cdot)$ to project all merged visual tokens $X_{\text{merged}} = \{\mathbf{X}_{\text{long}}, \mathbf{X}_{\text{short}}, \mathbf{X}_{\text{curr}}\}$ into visual observation tokens that are compatible with the LLM's input representation space:

$$\mathbf{E}_T^V = P_V(\mathbf{X}_{\text{merged}}), \quad (7)$$

where the $P_V(\cdot)$ is implemented as a two-layer MLP [54] and optimized in an end-to-end training manner. For instruction encoding, we use the off-the-shelf language tokenizer and embeing layer of LLM (Vicuna-7B [20]) to encode navigation instruction into language observation tokens $\mathbf{E}_T^L$. Then we concatenate the visual observation tokens $\mathbf{E}_T^V$, a navigation task indicator $\langle NAV \rangle$ and language observation tokens $\mathbf{E}_T^V$ form the final input token sequence. Here, the navigation task indicator $\langle NAV \rangle$ is adopted by following [111, 67] for accelerating the specific task learning and obtaining consistent output format. Finally, the complete input token sequence is fed into the LLM to infer four action tokens $\{\mathbf{E}_T^A, \cdots, \mathbf{E}_{T+3}^A\}$, as described below. We include a discussion on the input token format in the Supplementary Material

> **Input:** $\{Long\_term\_tokens\}\{Shot\_term\_tokens\}$ $\{Current\_tokens\} <NAV> \{Instruction\}$
> **Output:** $<Action\_0><Action\_1><Action\_2>$ $<Action\_3>$

The action tokens belong to the discrete action set $\{\texttt{FORWARD}, \texttt{TURN-LEFT}, \texttt{TURN-RIGHT}, \texttt{STOP}\}$. Following the standard configuration in existing navigation settings [76, 106], the forward action corresponds to a movement of 25 cm, and the turning actions represent a $30°$ rotation. This configuration is consistent with all training navigation data (Sec. V). Empirically, we find that predicting the next four steps yields optimal performance, which encourages Uni-NaVid to forecast long-horizon action sequences while still considering sufficient observations for accurate prediction. This multi-step prediction also supports asynchronous deployment, enabling non-blocking navigation performance in the real world. Please see the Supplementary Material for detailed elaboration.

V. DATA COLLECTION AND TRAINING

To train Uni-NaVidfor mastering multi-navigation tasks, it is crucial to gather extensive and diverse navigation data across various tasks and environments. However, directly collecting large amounts of real-world navigation data can be prohibitively expensive. To address this challenge, we propose two key strategies for training Uni-NaVid: First, we collect
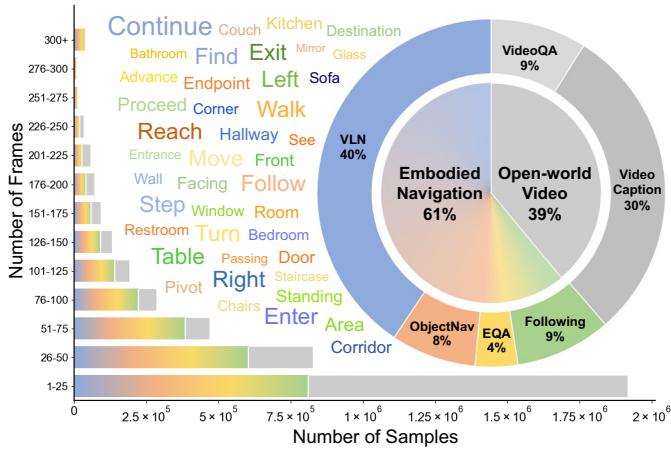
Fig. 3: **Visualization of training data.** We visualize the combination of training data (5.9M), video frame counts, and the most common words in navigation instructions.

Fig. 4: **Language-described human following benchmark.** We construct our human-following benchmark based on Habitat 3.0 [68] by incorporating textual descriptions for each avatar (eight in total, top row). The robot is required to comprehend these descriptions and accurately follow the designated individual in crowded environments.

multi-task navigation data from a wide range of synthetic environments (totaling 861 scenes) using a uniform input and output format, enabling Uni-NaVidto acquire general navigation skills. Second, we co-tune Uni-NaVidwith real-world video-based question-answering data, enhancing its ability to interpret real-world images and supporting its open-vocabulary knowledge acquisition.

A. Multi-Task Navigation Data.

We collect the largest multi-task navigation dataset to date within the Habitat simulator environment [76], comprising 3.6 million samples across four distinct navigation tasks, as described below. All tasks are curated within a unified framework. A detailed data collection strategy is provided in the Supplementary Materials.

(A) Vision-and-language navigation [44, 46] require the agent to interpret and ground instructions in visual observations, effectively combining linguistic and visual information to make sequential decisions. Specifically, the agent has to navigate based on landmarks and motions described in the text and stop nearby the correct destination. Here, we collect 2.4M navigation samples of mainstream VLN datasets, VLN-CE R2R [44] and RxR [47], that focus on continuous environments.

(B) Object Goal Navigation [76] involves an agent navigating an environment to locate a specific object based on provided visual or linguistic cues. This task evaluates the agent's ability to perceive objects, understand scene layout, and execute efficient search strategies. We collected 483k samples from datasets in the Habitat Matterport 3D dataset (HM3D ObjectNav) [70]. Note that, in HM3D ObjectNav, the agent is required to locate objects from a predefined category set (e.g., *sofa*, *chair*, and *bed*). Nevertheless, experiments demonstrate that our method generalizes to SOTA-level open-vocabulary object goal searching, as shown in Table V.

(C) Embodied question answering [90] requires the agent to navigate to the related area for question answering. It involves

spatial reasoning, object description, and understanding contextual information, requiring the ability to integrate perception, language comprehension, and decision-making. Following the setup in main stream EQA methods [21, 90], the agent first navigates to the target related to the question, issues a stop action, and then provides an answer. We collect 240k video-action samples and 10k video-answering samples on the MP3D-EQA dataset [21] on Matterport 3D environments [11].

(D) Human following [37, 25] requires the agent to track and follow a human target with a specific description in dynamic and crowded environments, *e.g.*, *"Follow the man in the blue t-shirt."*. The agent must recognize the appearance of the human, follow the correct person described in the instructions, predict their movement trajectory, and keep an appropriate distance while avoiding obstacles.

However, there is currently no human-following dataset that supports language-described human following in crowded environments (multi-person scenarios). To this end, we extend the Habitat 3.0 social navigation benchmark [68] by (1) adding textual descriptions for each avatar (8 in total, as illustrated in Fig. 4), (2) introducing additional distracting human avatars to simulate challenging real-world environments, and (3) deploying the robot and humans in the Habitat Matterport 3D dataset [101], which offers photo-realistic rendering quality and diverse large-scale scenes. The robot and target human are initialized nearby (using the same setting as [68]), with randomly moving distracting human avatars. Based on this setup, we collected 544k human-following navigation samples. We also add a detailed description in Supplementary Material. *This benchmark will also be released to benefit the navigation community.*

Unified navigation samples. The data statistics are presented in Figure 3. It is worth noting that the number of samples in VLN is relatively larger compared to other tasks. This is because VLN [44, 46] requires the agent to navigate all

landmarks described in the instructions, which often results in longer trajectories and, consequently more video-action samples. Here, we collect all navigation samples in a uniform format, including an egocentric RGB video, a natural language instruction, and four corresponding future actions. All data were collected from synthetic scenes across the Habitat-Matterport 3D (HM3D) and Matterport 3D (MP3D) datasets. We use the default settings of each environment, with a height range of 0.88 m to 1.25 m and a robot radius between 0.1 m and 0.6 m. This approach helps prevent overfitting to a specific robot embodiment. This approach helps prevent overfitting to a specific robot embodiment. Note that while there exist insightful techniques [24, 29] investigating navigation for robots of general sizes, our focus is primarily on uniform multi-task navigation.

B. Training Strategy of Uni-NaVid

Joint training on synthetic and real-world data. Although we collect navigation data from various environments, the diversity in both observations and instructions remains limited to a specific set of synthetic environments. To incorporate open-world knowledge, we follow previous Vision-and-Language Action models [111, 9], integrating open-world video question-answering during training. Specifically, we adopt a two-stage training process (a common strategy in Vision-and-Language models [54, 51, 80]): (1) First, we exclusively train the cross-modality projector (Equ. 7) using the same modality alignment dataset as LLaMA-VID [51]. (2) Second, we fine-tune both the projector and the Large Language Model (LLM) using 2.3M video question-answering data from publicly available datasets [7, 19, 51], along with 3.6M multi-task navigation samples. During training, we apply the online token merging to both the VQA samples and navigation samples, the only difference is the VAQ samples do not include navigation task indicator ⟨*NAV*⟩.

Training configuration. Uni-NaVid is trained on a cluster server with 40 NVIDIA H800 GPUs for approximately 35 hours, totaling 1400 GPU hours. For video data, we sample frames at 1 FPS to remove redundant information between consecutive frames. During training, the vision encoder (EVA-CLIP [82]) and large language model (Vicuna-7B [20]) are pre-loaded with default pre-trained weight. Following the training strategy of VLM [54], we optimize the trainable parameters for only 1 epoch.

VI. EXPERIMENT

We conduct experiments to evaluate Uni-NaVid on three specific aspects: (1) How does Uni-NaVid perform on individual tasks? (2) Does learning multiple navigation tasks lead to synergistic improvements? (3) Is the key design of our method effective? To evaluate the general-purpose navigation method, we conduct extensive experiments on individual navigation tasks, employing corresponding strong baselines. Additional details are provided in the supplemental material.

Benchmarks. We evaluate our method on various benchmarks across different navigation tasks. Given the diversity of benchmarks spanning various environments and simulators, we meticulously verify the scene splits to ensure no overlap exists between the training and validation scenes across benchmarks.

- *Vision-and-language navigation*: We test our method on the validation splits of the VLN-CE R2R [44] and RxR [46] benchmarks.
- *Object goal navigation*: We use the validation split of the Habitat Matterport 3D (HM3D) dataset [70], which requires the agent to find target objects from six categories (sofa, chair, TV, bed, toilet, and plant) in unseen environments. Moreover, to test generalizability, we also evaluate our method on the HM3D-OVON dataset [101], an open-vocabulary object navigation benchmark, in a zero-shot manner.
- *Embodied question-answering*: We use the validation split of the MP3D-EQA benchmark [90]. Additionally, we conduct experiments on the more recent Embodied Video Question Answering benchmark, OpenEQA [63].
- *Human following*: We evaluate our method alongside mainstream approaches on our proposed language-described human following benchmark.
- *Video understanding*: We follow the evaluation procedures of existing VQA methods [51]. We choose the ScanQA [7], MSVD [13], MSRVTT [96], and ActivityNet [10] datasets.

Metrics. To evaluate navigation performance, we follow the standard evaluation metrics [4], including success rate (SR), oracle success rate (OS), success weighted by path length (SPL) [3], trajectory length (TL), following rate (FR) [68], collision rate (CR) [68] and navigation error from goal (NE). Note that the success criteria change among different navigation tasks, we therefore use the default success criteria of each benchmark. For video understanding evaluation, we employ widely used metrics following existing works [7, 51].

A. Deployment Details of Uni-Navid.

Benchmark evaluation. For each navigation task, we adhere to the default settings of each navigation task [44, 76, 21, 37]. All tasks take an online captured RGB video (capturing one frame after each action) and a textual instruction as inputs, and output the next four actions (Sec. IV-C). The robot then executes the predicted actions and calls `STOP` once the first predicted action is a stop action. For VLN and EQA tasks, we directly use the text instruction provided by the benchmark episodes. For human following and object goal navigation, we transform the target information into an instruction by adding prefixes such as "Search for" or "Follow." Further details can be found in the supplemental material.

It is worth noting that for EQA [21] task, the agent executes navigation actions until a stop command is issued. We then remove the navigation-specific token `<NAV>` and query the questions using the navigation history. This strategy alleviates the ambiguity for the LLM in deciding whether to navigate or answer a question (See Table X).

Real-world deployment. For real-world deployment, we utilize a remote server with an NVIDIA A100 GPU to run

| Method | Observation | | | | VLN-CE R2R Val-Unseen | | | | |
	Pan.	Odom.	Depth	S.RGB	TL	NE↓	OS↑	SR↑	SPL↑
HPN+DN* [45]	✓	✓	✓		7.62	6.31	40.0	36.0	34.0
CMA* [31]	✓	✓	✓		10.90	6.20	52.0	41.0	36.0
VLN↻BERT*† [31]	✓	✓	✓		12.23	5.74	53.0	44.0	39.0
Sim2Sim* [43]	✓	✓	✓		10.69	6.07	52.0	43.0	36.0
GridMM* [87]	✓	✓	✓		13.36	5.11	61.0	49.0	41.0
HAMT*‡ [89]	✓	✓	✓		–	4.80	–	55.0	51.0
ETPNav* [1]	✓	✓	✓		11.99	4.71	65.0	57.0	49.0
InstructNav [61]	✓	✓	✓	✓	7.74	6.89	-	31.0	24.0
AG-CMTP [15]	✓	✓	✓		–	7.90	39.2	23.1	19.1
R2R-CMTP [15]	✓	✓	✓		–	7.90	38.0	26.4	22.7
LAW [73]		✓	✓	✓	8.89	6.83	44.0	35.0	31.0
CM2 [26]		✓	✓	✓	11.54	7.02	41.5	34.3	27.6
WS-MGMap [16]		✓	✓	✓	10.00	6.28	47.6	38.9	34.3
ETPNav.FF [88]		✓	✓	✓	-	5.95	**55.8**	44.9	30.4
Seq2Seq [44]			✓	✓	9.30	7.77	37.0	25.0	22.0
CMA [44]		✓		✓	8.64	7.37	40.0	32.0	30.0
NaVid [111]				✓	7.63	5.47	49.1	37.4	35.9
Uni-NaVid				✓	9.71	**5.58**	53.3	**47.0**	**42.7**

TABLE II: **Vision-and-language navigation (R2R).** Comparison on VLN-CE R2R [44] Val-Unseen. *: Methods use high-level action space. †: Methods use the same waypoint predictor proposed in [31]. ‡: Methods use additional visual data than MP3D scenes [11].

| Method | Observation | | | VLN-CE RxR Val-Unseen | | | | |
	Odom.	Depth	S.RGB	TL	NE↓	OS↑	SR↑	SPL↑
LAW* [73]	✓	✓	✓	4.01	10.87	21.0	8.0	8.0
CM2* [26]	✓	✓	✓	12.29	8.98	25.3	14.4	9.2
WS-MGMap* [16]	✓	✓	✓	10.80	9.83	29.8	15.0	12.1
ETPNav.FF [88]	✓	✓	✓	-	8.79	36.7	25.5	18.1
Seq2Seq* [44]		✓	✓	1.16	11.8	5.02	3.51	3.43
CMA* [44]	✓		✓	5.09	11.7	10.7	4.41	2.47
A²Nav† [17]			✓	–	–	–	16.8	6.3
NaVid* [111]			✓	10.59	8.41	34.5	23.8	21.2
Uni-NaVid			✓	15.8	**6.24**	**55.5**	**48.7**	**40.9**

TABLE III: **Vision-and-language navigation (RxR).** Comparison on VLN-CE RxR [47] Val-Unseen. *: only trained on VLN-CE R2R.

| Method | Observation | | | HM3D ObjectNav | |
	Odom.	Depth	S.RGB	SR↑	SPL↑
DD-PPO [91]	✓	✓	✓	27.9	14.2
Habitat-Web [71]	✓	✓	✓	57.6	23.8
InstructNav [61]	✓	✓	✓	58.0	20.9
PIRLNav-IL [72]	✓		✓	64.1	27.1
PIRLNav-IL-RL [72]	✓		✓	70.4	34.1
OVRL [99]	✓		✓	62.0	26.8
OVRL-v2 [98]	✓		✓	64.7	28.1
Uni-NaVid			✓	**73.7**	**37.1**

TABLE IV: **Object goal navigation.** Comparison on Habitat Matterport 3D [70] ObjectNav dataset.

| Method | VAL SEEN | | VAL SEEN SYNONYMS | | VAL UNSEEN | |
	SR↑	SPL↑	SR↑	SPL↑	SR↑	SPL↑
BC	11.1	4.5	9.9	3.8	5.4	1.9
DAgger	11.1	4.5	9.9	3.8	5.4	1.9
RL	18.1	9.4	15.0	7.4	10.2	4.7
BCRL	39.2	18.7	27.8	11.7	18.6	7.5
DAgRL	**41.3**	**21.2**	29.4	14.4	18.3	7.9
VLFM* [100]	35.2	18.6	32.4	17.3	35.2	19.6
DAgRL+OD [101]	38.5	21.1	39.0	21.4	37.1	**19.8**
Uni-NaVid*	**41.3**	21.1	**43.9**	**21.8**	**39.5**	19.8

TABLE V: **Object goal navigation.** Comparison on HM3D-OVON [101]. * : denotes zero-shot methods.

and A²Nav [17]) that use ChatGPT with only text inputs for visual language navigation (VLN), these approaches often face challenges in transitioning between text prompts and visual information, resulting in less than satisfactory outcomes. Furthermore, it is important to note that the trajectories in RxR are more diverse and involve longer paths with detailed landmark descriptions, making RxR widely regarded as more challenging than R2R. However, our method achieves consistent performance across both R2R and RxR, with slightly better results on RxR (+3.6 SR(%)), demonstrating its ability to effectively leverage detailed instructions to navigate diverse trajectories. We add experiments of removing RxR samples in Supplemntal Material, where our method still achive STOA performance (+23.9 SR(%)) against NaVid.

Comparison on object goal navigation. We conduct the experiments on HM3D [70] to compare Uni-NaVid with mainstream methods [91, 71, 72, 99, 98] that also learn from ObjectNav data. The results, shown in Table IV, demonstrate that our approach achieves the best performance. Note that methods not utilizing odometry face challenges as they must rely on implicit memory to retain the historical trajectory. Nevertheless, Uni-NaVid still achieves significant gains in SR (+4.7%) and SPL (+8.8%) compared to previous state-of-the-art methods. Additionally, we believe our method's ObjectNav performance can be further enhanced by incorporating reinforcement learning techniques, as demonstrated by PIRLNav [72] and Poliformer [106].

To evaluate the generalization ability for open-vocabulary objects, we evaluate our method on the open-vocabulary object goal navigation benchmark (HM3D-OVON [101]) in a zero-

Uni-NaVid, which processes observations (along with text instructions) and sends commands to a local robot to execute the predicted actions. Uni-NaVid requires approximately 0.2 seconds to generate the next four actions. During navigation, the robot asynchronously compresses and uploads the latest observations to the model while executing pending actions. Refer to the supplementary video for real-world navigation performance.

B. Individual Task Results

Comparison on vision-and-language navigation. We evaluate our method with mainstream baselines on two publicly available benchmarks: VLN-CE R2R [44] and RxR [47]. The results are shown in Table II and Table III. We find that our methods achieve SOTA-level performance on both datasets using only RGB videos as observations. In comparison to NaVid [111], which is also a vision language model that is solely trained on VLN data, our approach demonstrates significant improvements, with a +25.7% increase in Success Rate (SR) on R2R. For zero-shot methods (InstructNav [61]

Method	Action Type			MP3D EQA
	D.E.	C.E.	GT	**ACC↑**
NaviLLM [114]	✓			44.5
Uni-NaVid		✓		**47.3**
EQA(habitat-lab) [21]			✓	46.0
NaviLLM [114]			✓	47.4
Uni-NaVid			✓	**54.4**

TABLE VI: **Embodied question answering.** Comparison on Habitat Matterport3D EQA dataset [21].

Method	Observation		Human Following Dataset		
	H.Det.	S.RGB	**SR↑**	**FR↑**	**CR↓**
PoliFormer [106]		✓	2.79	20.35	2.93
PoliFormer* [106]	✓	✓	14.67	37.14	4.29
PoliFormer† [106]	✓	✓	25.29	47.16	6.78
IBVS* [28]	✓	✓	46.08	62.64	0.84
IBVS† [28]	✓	✓	50.58	68.89	**0.80**
Uni-NaVid		✓	**61.21**	**71.93**	2.07

TABLE VII: **Human following.** Comparison on Human Following Dataset. *: Methods use GroundingDINO [58] as the open-vocabulary human detector. †: Methods use the ground-truth bounding box provided by the simulator.

Method	ScanQA				
	EM ↑	BLUE-1 ↑	ROUGE ↑	METEOR ↑	CIDEr ↑
V.N.+MCAN [105]	19.71	29.46	30.97	12.07	58.23
S.R.+MCAN [105]	20.56	27.85	30.68	11.97	57.56
3D-LLM(flamingo) [32]	23.2	32.60	34.80	13.5	65.6
NaviLLM [114]	26.27	39.73	40.23	16.56	80.77
BridgeQA [65]	**31.29**	34.49	43.26	16.51	83.75
Uni-NaVid	28.01	**46.85**	**45.74**	**19.24**	**94.72**

TABLE VIII: **Embodied video question answering.** Comparison on ScanQA [7] benchmark.

Method	MSVD-QA		MSRVTT-QA		ActivityNet-QA	
	Acc↑	Score↑	Acc↑	Score↑	Acc↑	Score↑
VideoLLaMA [107]	51.6	2.5	29.6	1.8	12.4	1.1
VideoChat [49]	56.3	2.8	45.0	2.5	26.5	2.2
VideoChatGPT [62]	64.9	3.3	49.3	2.8	35.2	2.7
BT-Adapter [56]	67.5	3.7	57.0	3.2	45.7	3.2
Chat-UniVi [39]	65.0	3.6	54.6	3.1	45.8	3.2
LLaMA-VID [51]	69.7	3.7	57.7	3.2	47.4	3.3
VideoChat2 [50]	70.0	**3.9**	54.1	3.3	49.1	3.3
Video-LLaVA [53]	70.7	**3.9**	59.2	**3.5**	45.3	3.3
ST-LLM [57]	**74.6**	**3.9**	**63.2**	3.4	50.9	3.3
Uni-NaVid	69.6	**3.9**	59.3	**3.5**	**51.4**	**3.7**

TABLE IX: **Video question answering.** Comparison with leading methods (all based on Vicuna-7B [20]) on VQA benchamarks.

shot manner. The results in Table V demonstrate that our method achieves significant improvement over the zero-shot method (VLFM [100]) and even outperforms the fine-tuned method (DAgRL+OD [101]) on the VAL SEEN and VAL UNSEEN splits. This proves the generalizability of our method.

Comparison on embodied question answering. The evaluation results on MP3D-EQA [90] are presented in Table VI. Despite navigating in continuous environments (CE), our method outperforms existing approaches (*e.g.,* NaviLLM [114] leverage the same evaluation strategy in Sec. VI-A) that operate within discrete landmark-based environments (DE). Moreover, when provided with the ground truth (GT) navigation trajectory, our method shows a significant improvement, demonstrating its ability to understand navigation history effectively. We also report our performance on the more challenging EM-EQA benchmark, OpenEQA [63], in the Supplemental Material, which includes more complex questions. Our method achieves comparable performance to GPT-4V with scene captions [63].

Comparison on human following. We compared our method with two most relative methods PoliFormer [106] and IBVS [28]. Since both methods require a specific human bounding box as input, obtained from an upstream algorithm, we use the bounding box from the open-world object detector GroundingDINO [58] and the ground truth provided by the simulator to evaluate the human following performance of the comparison methods under various setups. As shown in Table VII, Uni-NaVid outperforms the comparison methods on both SR (+21.0%) and FR (+4.4%) while maintaining low CR under any setup, even when they use ground truth bounding boxes as input. This demonstrates that Uni-NaVid can effectively infer instructions and follow the correct human, as

well as predict the human's movement patterns accurately. We include additional human-following experiments in various environments, such as HSSD [42] and MP3D [11], in the Supplemental Material. Our method consistently demonstrates SOTA performance across these settings.

Comparison on video question answering. We first evaluate our method on ScanQA [7] on Tab. VIII. Compared to mainstream baselines, we find that Uni-NaVid archives the best performance on four metrics, including BLUE-1 (+17.9%), ROUGE (+5.7%), METEOR (+16.2%), and CIDEr (+13.1%). This proves the superiority of our methods on spatial scene understanding. Note that the EM metric requires an exact match between the question and answer, which is not well-suited to our method, as it is designed to learn from diverse data and generate flexible responses.

We further evaluate our method on open-ended video question-answering benchmarks [13, 96, 10], as presented in Table IX. To ensure a fair comparison, we focus on methods that employ the same large language model backbone (Vicuna-7B [20]). The results indicate that even after extensive token merging (Sec. IV-B), Uni-NaVidachieves performance comparable to state-of-the-art methods. This demonstrates the effectiveness of both our token merging and training strategies, while also highlighting robust open-world understanding capabilities.

C. Qualitative Results in Real-World

We conducted extensive experiments on real-world environments (experiment details are provided in the supplemental
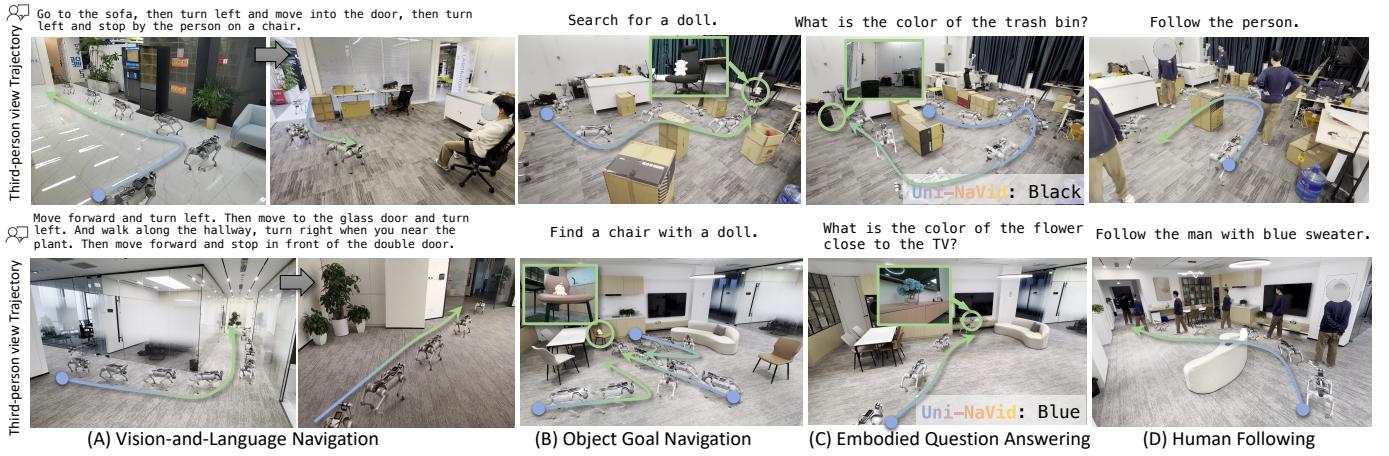
Fig. 5: **Visual results of Uni-NaVid in real-world.** We deploy Uni-NaVid across diverse environments to execute instructions in a zero-shot setting. We provide third-person views with robot's trajectory, showing effective navigation performance. We indicate the starting point as a blue dot and the ending point as a green arrow.
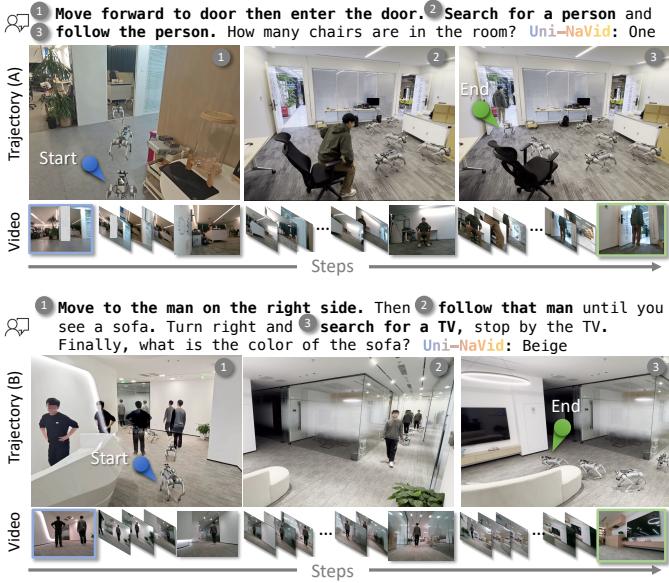

Fig. 6: **Vivusal results of Uni-NaVid on compostional tasks.** The agent is required to execute complex instructions involving multiple navigation tasks. Our method successfully accomplishes these navigation tasks sequentially. Notably, both the instructions and environments are novel to our approach. Please refer to the supplementary videos.

material) under diverse environments in a zero-shot manner. Notably, both the instructions and environments are novel to our method. We first evaluated the performance of individual navigation tasks (Fig. 5), including (A) vision-and-language navigation, (B) object goal navigation, (C) embodied question answering, and (D) human following. We found that Uni-NaVid can understand diverse instructions and demonstrates impressive performance in long-horizon navigation tasks (e.g., navigating

across hallways and entering rooms), as well as in searching for out-of-vision objects and answering subsequent questions. Moreover, the agent is capable of following a human even when the person's appearance deviates from the description of the avatar in the human-following dataset (Sec. V-A). The statistics of the corresponding real-world experiments can be found in the Supplemental Material.

In addition to individual navigation tasks, we also evaluate our method on more complex instructions involving multiple navigation tasks (Fig. 6). In this scenario, the agent is required to sequentially execute the navigation tasks described in the language instructions. Our model demonstrates impressiove performance in aligning the current navigation process with the instructions to reason about the current state of navigation. Furthermore, we provide a detailed illustration of action prediction during navigation in Fig. 7, where we plot the predicted action probabilities of Uni-NaVid. Notably, with only slight differences in object descriptions, *e.g., 'chair with a toy'* and *'chair with a sweater'*. Specifically, our method successfully distinguishes between the locations and predicts actions accordingly. Interestingly, the action probabilities (for the next four actions) reveal a sequential order of actions: first turning right/left, followed by moving forward. We provide additional visual results of our method in the Supplemental Material and encourage the audience to view our video, which showcases the real-world performance of our method.

D. Ablation Study

Visualization of training strategy. We present a visualization of the training strategy's performance in Figure 8. In Fig. 8 (a), we compare training on a single navigation task with training across multiple tasks. The results demonstrate the synergistic benefits of multi-task learning, which yields consistent performance improvements across all navigation tasks. Notably, VLN, ObjectNav, and EQA exhibit more significant improvements, while Following shows relatively
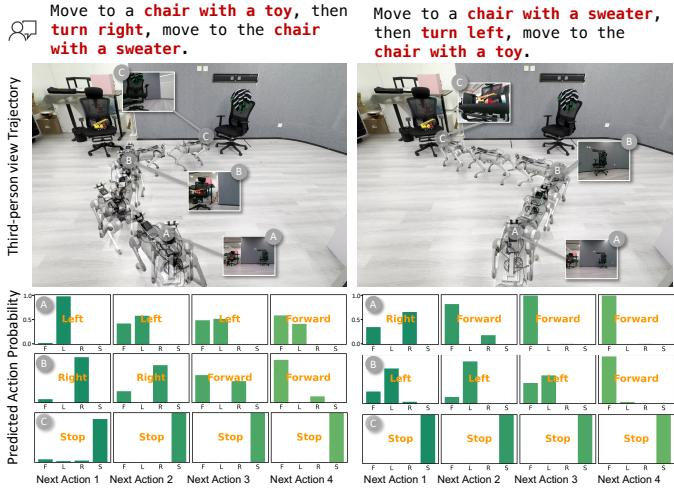
Fig. 7: **Action prediction on the VLN tasks.** We evaluate Uni-NaVid on challenging open-vocabulary objects, requiring it to recognize the target objects and follow the specified motions. We provide the predicted action probabilities (for the next four actions) to demonstrate its break-in navigation capability.
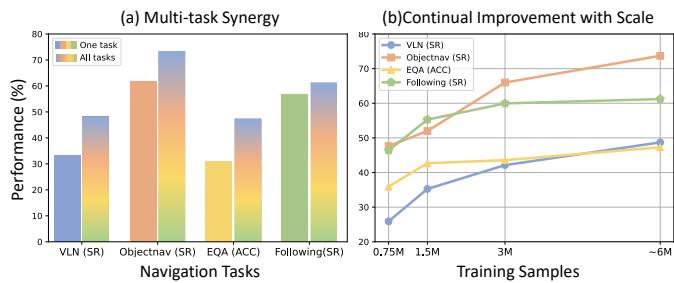

Fig. 8: **Comparsion on multi-task training and data scale.** (a) We present the multi-task synergy of our method, illustrating the performance comparison between training with a single task and training with multiple tasks; (b) we demonstrate the performance across different navigation tasks under varying numbers of training samples.

smaller gains. We attribute this difference to the lower reliance of the Following task on historical context. Additionally, we investigate the influence of data scale on navigation performance (Figure 8 (b)). We observe that performance improves across all navigation tasks with larger data volumes. However, the incremental gain diminishes (from 3M to 6M samples), potentially due to limitations in the data diversity of simulators. Specifically, for the Following task, the reason for the slower convergence is the heavy occlusion caused by obstacles or other humans. This highlights the need for more high-quality following data samples, which can enable our model to learn more effectively and perform better in highly dynamic environments.

Ablation on training strategy and architecture. We conduct experiments to evaluate the effectiveness of the training

Type	VLN (SR↑)	ObjNav (SR↑)	EQA (ACC↑)	Follow (SR↑)
No <Nav> token	35.2	69.1	20.4	55.1
No VQA data	40.5	50.6	1.19	58.8
Curr.	9.61	44.3	32.5	56.3
Curr.+Short.	39.7	67.8	44.1	59.7
Curr.+Short.+Long.	**48.7**	**73.7**	**47.3**	**61.2**

TABLE X: **Ablation study on training strategy and architecture**. For each ablation type, we retrain the entire model and evaluate its performance across four navigation tasks.

strategy and token merging designs (Tab. X). Our results indicate that the absence of <NAV> and VQA data leads to a performance decline across all tasks, similar findings can be found in [14, 111]. Notably, the performance drop is most obviously in EQA, as the lack of <NAV> special token makes the model misinterpret whether it should answer questions or output actions. Additionally, without VQA data, the agent's ability to answer questions drops significantly, almost rendering it incapable of correctly answering questions. We believe this is due to the catastrophic forgetting problem in LLMs, where the model loses open-world knowledge by being trained solely on navigation-related data.

From the performance of different memory designs, we find that both short-term and long-term memory visual tokens contribute to performance improvements. In particular, the VLN task shows the most significant performance drop (-80.3% SR) when visual memory is removed, as the lack of memory hinders the alignment of visual history with instructions. For the Following task, the absence of memory results in only a minor performance decline (-8% SR), as this task primarily relies on recent frames to track the target. Additional ablation studies on architecture and hyperparameters are provided in the Supplementary Material.

VII. LIMITATIONS

Despite the promising results, Uni-NaVid has several limitations. *First*, Uni-NaVid is trained and evaluated on four well-defined navigation tasks, while there exists a large body of literature on insightful and practical navigation datasets [86, 112]. We believe that collecting data from these datasets could further enhance the navigation capabilities of our method. *Second*, our method is designed to acquire multi-task navigation capabilities under the assumption that the robot is of standard size (see Section V-A). To extend it to robots of general sizes, a convincing approach is to incorporate prior knowledge of the robot's size, as demonstrated in [24, 29]. *Third*, our method is currently limited to predicting simple trajectories composed of a short horizon of future low-level discrete actions. This limitation could be alleviated by extending the moel to predict continuous and smooth trajectories with techniques from motion planning [79, 81] or autonomous driving [18, 52].

VIII. DISCUSSION AND CONCLUSION

In this paper, we introduce an efficient vision-language-action (VLA) model, Uni-NaVid, designed to acquire general embodied navigation skills through learning multi-task navigation

data. To efficiently encode the online-captured video sequences during navigation, we develop an online visual token merging mechanism that separately processes current observations, short-term observations, and long-term observations. This design enables our approach to operate at an average speed of 5 Hz. We also collect 3.6 million navigation data points across four highly demanded embodied navigation tasks, including vision-and-language navigation, object goal navigation, embodied question answering, and human following. Extensive experiments and ablation studies demonstrate that our method achieves SOTA-level performance using only monocular videos as input, highlighting our model's superior capability in learning multiple navigation tasks. Moreover, we deploy Uni-NaVid in real-world environments, demonstrating impressive generalizability and versatile navigation performance in real worlds.

Future works. Our work serves merely as a starting point of general-purpose navigation, and we hope it will inspire future directions in this field:

- *Benchmarking.* With the consistent development of embodied navigation, there is a growing need for general-purpose navigation benchmarking. Such a benchmark would help researchers better position their work and drive progress in the navigation community.
- *Architecture.* We would like to further enhance the practicality of our architecture by tackling very long-horizon tasks (e.g., navigating across buildings) and incorporating advanced motion planning techniques [81, 18].
- *Application.* We would like to apply our method to applications such as robotic guide dogs and home service robots. Additionally, we are excited to extend this technique to other embodied AI tasks, such as mobile manipulation [110, 92, 55].

REFERENCES

[1] Dong An, Hanqing Wang, Wenguan Wang, Zun Wang, Yan Huang, Keji He, and Liang Wang. Etpnav: Evolving topological planning for vision-language navigation in continuous environments. *arXiv preprint arXiv:2304.03047*, 2023.

[2] Peter Anderson, Angel Chang, Devendra Singh Chaplot, Alexey Dosovitskiy, Saurabh Gupta, Vladlen Koltun, Jana Kosecka, Jitendra Malik, Roozbeh Mottaghi, Manolis Savva, et al. On evaluation of embodied navigation agents. *arXiv preprint arXiv:1807.06757*, 2018.

[3] Peter Anderson, Angel Chang, Devendra Singh Chaplot, Alexey Dosovitskiy, Saurabh Gupta, Vladlen Koltun, Jana Kosecka, Jitendra Malik, Roozbeh Mottaghi, Manolis Savva, et al. On evaluation of embodied navigation agents. *arXiv preprint arXiv:1807.06757*, 2018.

[4] 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. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3674–3683, 2018.

[5] Peter Anderson, Ayush Shrivastava, Joanne Truong, Arjun Majumdar, Devi Parikh, Dhruv Batra, and Stefan Lee. Sim-to-real transfer for vision-and-language navigation. In *Conference on Robot Learning*, pages 671–681. PMLR, 2021.

[6] RC Atkinson and RM Shiffrin. Human memory: A proposed system and its control processes (vol. 2). *The Psychology of Learning and Motivation: Advances in Research and Theory*, pages 89–195, 1968.

[7] 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*, pages 19129–19139, 2022.

[8] Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your vit but faster. *ArXiv*, abs/2210.09461, 2022.

[9] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*, 2023.

[10] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *Proceedings of the ieee conference on computer vision and pattern recognition*, pages 961–970, 2015.

[11] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niebner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from rgb-d data in indoor environments. In *2017 International Conference on 3D Vision (3DV)*, pages 667–676. IEEE, 2017.

[12] Devendra Singh Chaplot, Dhiraj Prakashchand Gandhi, Abhinav Gupta, and Russ R Salakhutdinov. Object goal navigation using goal-oriented semantic exploration. *Advances in Neural Information Processing Systems*, 33: 4247–4258, 2020.

[13] David Chen and William B Dolan. Collecting highly parallel data for paraphrase evaluation. In *Proceedings of the 49th annual meeting of the association for computational linguistics: human language technologies*, pages 190–200, 2011.

[14] Jun Chen, Deyao Zhu, Xiaoqian Shen, Xiang Li, Zechun Liu, Pengchuan Zhang, Raghuraman Krishnamoorthi, Vikas Chandra, Yunyang Xiong, and Mohamed Elhoseiny. Minigpt-v2: large language model as a unified interface for vision-language multi-task learning. *arXiv preprint arXiv:2310.09478*, 2023.

[15] Kevin Chen, Junshen K Chen, Jo Chuang, Marynel Vázquez, and Silvio Savarese. Topological planning with transformers for vision-and-language navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11276–11286,

2021.

[16] Peihao Chen, Dongyu Ji, Kunyang Lin, Runhao Zeng, Thomas H Li, Mingkui Tan, and Chuang Gan. Weakly-supervised multi-granularity map learning for vision-and-language navigation. *arXiv preprint arXiv:2210.07506*, 2022.

[17] Peihao Chen, Xinyu Sun, Hongyan Zhi, Runhao Zeng, Thomas H Li, Gaowen Liu, Mingkui Tan, and Chuang Gan. Action-aware zero-shot robot navigation by exploiting vision-and-language ability of foundation models. *arXiv preprint arXiv:2308.07997*, 2023.

[18] Shaoyu Chen, Bo Jiang, Hao Gao, Bencheng Liao, Qing Xu, Qian Zhang, Chang Huang, Wenyu Liu, and Xinggang Wang. Vadv2: End-to-end vectorized autonomous driving via probabilistic planning. *arXiv preprint arXiv:2402.13243*, 2024.

[19] Tsai-Shien Chen, Aliaksandr Siarohin, Willi Menapace, Ekaterina Deyneka, Hsiang-wei Chao, Byung Eun Jeon, Yuwei Fang, Hsin-Ying Lee, Jian Ren, Ming-Hsuan Yang, et al. Panda-70m: Captioning 70m videos with multiple cross-modality teachers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13320–13331, 2024.

[20] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. *See https://vicuna. lmsys. org (accessed 14 April 2023)*, 2023.

[21] Abhishek Das, Samyak Datta, Georgia Gkioxari, Stefan Lee, Devi Parikh, and Dhruv Batra. Embodied question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–10, 2018.

[22] Vishnu Sashank Dorbala, Gunnar Sigurdsson, Robinson Piramuthu, Jesse Thomason, and Gaurav S Sukhatme. Clip-nav: Using clip for zero-shot vision-and-language navigation. *arXiv preprint arXiv:2211.16649*, 2022.

[23] Jiafei Duan, Samson Yu, Hui Li Tan, Hongyuan Zhu, and Cheston Tan. A survey of embodied ai: From simulators to research tasks. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 6(2):230–244, 2022.

[24] Ainaz Eftekhar, Luca Weihs, Rose Hendrix, Ege Caglar, Jordi Salvador, Alvaro Herrasti, Winson Han, Eli VanderBil, Aniruddha Kembhavi, Ali Farhadi, et al. The one ring: a robotic indoor navigation generalist. *arXiv preprint arXiv:2412.14401*, 2024.

[25] Anthony Francis, Claudia Pérez-d'Arpino, Chengshu Li, Fei Xia, Alexandre Alahi, Rachid Alami, Aniket Bera, Abhijat Biswas, Joydeep Biswas, Rohan Chandra, et al. Principles and guidelines for evaluating social robot navigation algorithms. *arXiv preprint arXiv:2306.16740*, 2023.

[26] Georgios Georgakis, Karl Schmeckpeper, Karan Wanchoo, Soham Dan, Eleni Miltsakaki, Dan Roth, and Kostas Daniilidis. Cross-modal map learning for vision and language navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15460–15470, 2022.

[27] Theophile Gervet, Soumith Chintala, Dhruv Batra, Jitendra Malik, and Devendra Singh Chaplot. Navigating to objects in the real world. *Science Robotics*, 8(79): eadf6991, 2023.

[28] Meenakshi Gupta, Swagat Kumar, Laxmidhar Behera, and Venkatesh K Subramanian. A novel vision-based tracking algorithm for a human-following mobile robot. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 47(7):1415–1427, 2016.

[29] Noriaki Hirose, Dhruv Shah, Ajay Sridhar, and Sergey Levine. Exaug: Robot-conditioned navigation policies via geometric experience augmentation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4077–4084. IEEE, 2023.

[30] Wenyi Hong, Weihan Wang, Ming Ding, Wenmeng Yu, Qingsong Lv, Yan Wang, Yean Cheng, Shiyu Huang, Junhui Ji, Zhao Xue, Lei Zhao, Zhuoyi Yang, Xiaotao Gu, Xiaohan Zhang, Guanyu Feng, Da Yin, Zihan Wang, Ji Qi, Xixuan Song, Peng Zhang, De-Feng Liu, Bin Xu, Juanzi Li, Yu-Chen Dong, and Jie Tang. Cogvlm2: Visual language models for image and video understanding. *ArXiv*, abs/2408.16500, 2024. URL https://api.semanticscholar.org/CorpusID:272146264.

[31] Yicong Hong, Zun Wang, Qi Wu, and Stephen Gould. Bridging the gap between learning in discrete and continuous environments for vision-and-language navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15439–15449, 2022.

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

[33] Shanee S Honig, Tal Oron-Gilad, Hanan Zaichyk, Vardit Sarne-Fleischmann, Samuel Olatunji, and Yael Edan. Toward socially aware person-following robots. *IEEE Transactions on Cognitive and Developmental Systems*, 10(4):936–954, 2018.

[34] Chenguang Huang, Oier Mees, Andy Zeng, and Wolfram Burgard. Visual language maps for robot navigation. *arXiv preprint arXiv:2210.05714*, 2022.

[35] Chenguang Huang, Oier Mees, Andy Zeng, and Wolfram Burgard. Visual language maps for robot navigation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 10608–10615. IEEE, 2023.

[36] Yulong Huang, Yonggang Zhang, Peng Shi, Zhemin Wu, Junhui Qian, and Jonathon A Chambers. Robust kalman filters based on gaussian scale mixture distributions with application to target tracking. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 49(10):2082–2096, 2017.

[37] Md Jahidul Islam, Jungseok Hong, and Junaed Sattar.

Person-following by autonomous robots: A categorical overview. *The International Journal of Robotics Research*, 38(14):1581–1618, 2019.

[38] Md Mofijul Islam, Alexi Gladstone, Riashat Islam, and Tariq Iqbal. Eqa-mx: Embodied question answering using multimodal expression. In *The Twelfth International Conference on Learning Representations*, 2023.

[39] Peng Jin, Ryuichi Takanobu, Wancai Zhang, Xiaochun Cao, and Li Yuan. Chat-univi: Unified visual representation empowers large language models with image and video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13700–13710, 2024.

[40] Abhishek Kadian, Joanne Truong, Aaron Gokaslan, Alexander Clegg, Erik Wijmans, Stefan Lee, Manolis Savva, Sonia Chernova, and Dhruv Batra. Sim2real predictivity: Does evaluation in simulation predict real-world performance? *IEEE Robotics and Automation Letters*, 5(4):6670–6677, 2020.

[41] Linh Kästner, Bassel Fatloun, Zhengcheng Shen, Daniel Gawrisch, and Jens Lambrecht. Human-following and-guiding in crowded environments using semantic deep-reinforcement-learning for mobile service robots. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 833–839, 2022.

[42] Mukul Khanna, Yongsen Mao, Hanxiao Jiang, Sanjay Haresh, Brennan Shacklett, Dhruv Batra, Alexander Clegg, Eric Undersander, Angel X Chang, and Manolis Savva. Habitat synthetic scenes dataset (hssd-200): An analysis of 3d scene scale and realism tradeoffs for objectgoal navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16384–16393, 2024.

[43] Jacob Krantz and Stefan Lee. Sim-2-sim transfer for vision-and-language navigation in continuous environments. In *European Conference on Computer Vision*, pages 588–603. Springer, 2022.

[44] Jacob Krantz, Erik Wijmans, Arjun Majumdar, Dhruv Batra, and Stefan Lee. Beyond the nav-graph: Vision-and-language navigation in continuous environments. In *European Conference on Computer Vision*, 2020. URL https://api.semanticscholar.org/CorpusID:214802389.

[45] Jacob Krantz, Aaron Gokaslan, Dhruv Batra, Stefan Lee, and Oleksandr Maksymets. Waypoint models for instruction-guided navigation in continuous environments. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15162–15171, 2021.

[46] Alexander Ku, Peter Anderson, Roma Patel, Eugene Ie, and Jason Baldridge. Room-across-room: Multilingual vision-and-language navigation with dense spatiotemporal grounding. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4392–4412, 2020.

[47] Alexander Ku, Peter Anderson, Roma Patel, Eugene Ie, and Jason Baldridge. Room-across-room: Multilingual

vision-and-language navigation with dense spatiotemporal grounding. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4392–4412, 2020.

[48] Yuxuan Kuang, Hai Lin, and Meng Jiang. Openfm-nav: Towards open-set zero-shot object navigation via vision-language foundation models. *arXiv preprint arXiv:2402.10670*, 2024.

[49] KunChang Li, Yinan He, Yi Wang, Yizhuo Li, Wenhai Wang, Ping Luo, Yali Wang, Limin Wang, and Yu Qiao. Videochat: Chat-centric video understanding. *arXiv preprint arXiv:2305.06355*, 2023.

[50] Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Yi Liu, Zun Wang, Jilan Xu, Guo Chen, Ping Luo, et al. Mvbench: A comprehensive multi-modal video understanding benchmark. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22195–22206, 2024.

[51] Yanwei Li, Chengyao Wang, and Jiaya Jia. Llama-vid: An image is worth 2 tokens in large language models. *arXiv preprint arXiv:2311.17043*, 2023.

[52] Bencheng Liao, Shaoyu Chen, Haoran Yin, Bo Jiang, Cheng Wang, Sixu Yan, Xinbang Zhang, Xiangyu Li, Ying Zhang, Qian Zhang, et al. Diffusiondrive: Truncated diffusion model for end-to-end autonomous driving. *arXiv preprint arXiv:2411.15139*, 2024.

[53] Bin Lin, Yang Ye, Bin Zhu, Jiaxi Cui, Munan Ning, Peng Jin, and Li Yuan. Video-llava: Learning united visual representation by alignment before projection. *arXiv preprint arXiv:2311.10122*, 2023.

[54] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *NeurIPS*, 2023.

[55] Peiqi Liu, Yaswanth Orru, Chris Paxton, Nur Muhammad Mahi Shafiullah, and Lerrel Pinto. Ok-robot: What really matters in integrating open-knowledge models for robotics. *arXiv preprint arXiv:2401.12202*, 2024.

[56] Ruyang Liu, Chen Li, Yixiao Ge, Thomas H Li, Ying Shan, and Ge Li. Bt-adapter: Video conversation is feasible without video instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13658–13667, 2024.

[57] Ruyang Liu, Chen Li, Haoran Tang, Yixiao Ge, Ying Shan, and Ge Li. St-llm: Large language models are effective temporal learners. In *European Conference on Computer Vision*, pages 1–18. Springer, 2025.

[58] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499*, 2023.

[59] Yang Liu, Weixing Chen, Yongjie Bai, Xiaodan Liang, Guanbin Li, Wen Gao, and Liang Lin. Aligning cyber space with physical world: A comprehensive survey on embodied ai. *arXiv preprint arXiv:2407.06886*, 2024.

[60] Yuxing Long, Xiaoqi Li, Wenzhe Cai, and Hao Dong. Discuss before moving: Visual language nav-

igation via multi-expert discussions. *arXiv preprint arXiv:2309.11382*, 2023.

[61] Yuxing Long, Wenzhe Cai, Hongcheng Wang, Guanqi Zhan, and Hao Dong. Instructnav: Zero-shot system for generic instruction navigation in unexplored environment. *arXiv preprint arXiv:2406.04882*, 2024.

[62] Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Shahbaz Khan. Video-chatgpt: Towards detailed video understanding via large vision and language models. *arXiv preprint arXiv:2306.05424*, 2023.

[63] Arjun Majumdar, Anurag Ajay, Xiaohan Zhang, Pranav Putta, Sriram Yenamandra, Mikael Henaff, Sneha Silwal, Paul Mcvay, Oleksandr Maksymets, Sergio Arnaud, et al. Openeqa: Embodied question answering in the era of foundation models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16488–16498, 2024.

[64] Christoforos Mavrogiannis, Francesca Baldini, Allan Wang, Dapeng Zhao, Pete Trautman, Aaron Steinfeld, and Jean Oh. Core challenges of social robot navigation: A survey. *ACM Transactions on Human-Robot Interaction*, 12(3):1–39, 2023.

[65] Wentao Mo and Yang Liu. Bridging the gap between 2d and 3d visual question answering: A fusion approach for 3d vqa. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 4261–4268, 2024.

[66] Khanh Nguyen, Debadeepta Dey, Chris Brockett, and Bill Dolan. Vision-based navigation with language-based assistance via imitation learning with indirect intervention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12527–12537, 2019.

[67] OpenAI. Gpt-4 technical report, 2023.

[68] Xavier Puig, Eric Undersander, Andrew Szot, Mikael Dallaire Cote, Tsung-Yen Yang, Ruslan Partsey, Ruta Desai, Alexander William Clegg, Michal Hlavac, So Yeon Min, et al. Habitat 3.0: A co-habitat for humans, avatars and robots. *arXiv preprint arXiv:2310.13724*, 2023.

[69] 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.

[70] Santhosh Kumar Ramakrishnan, Aaron Gokaslan, Erik Wijmans, Oleksandr Maksymets, Alexander Clegg, John M 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. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021.

[71] Ram Ramrakhya, Eric Undersander, Dhruv Batra, and Abhishek Das. Habitat-web: Learning embodied object-search strategies from human demonstrations at scale. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5173–5183, 2022.

[72] Ram Ramrakhya, Dhruv Batra, Erik Wijmans, and Abhishek Das. Pirlnav: Pretraining with imitation and rl finetuning for objectnav. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17896–17906, 2023.

[73] Sonia Raychaudhuri, Saim Wani, Shivansh Patel, Unnat Jain, and Angel X Chang. Language-aligned waypoint (law) supervision for vision-and-language navigation in continuous environments. *arXiv preprint arXiv:2109.15207*, 2021.

[74] Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 627–635. JMLR Workshop and Conference Proceedings, 2011.

[75] Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, Devi Parikh, and Dhruv Batra. Habitat: A Platform for Embodied AI Research. *ICCV*, 2019.

[76] Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, et al. Habitat: A platform for embodied ai research. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9339–9347, 2019.

[77] James A Sethian. Fast marching methods. *SIAM review*, 41(2):199–235, 1999.

[78] Dhruv Shah, Błażej Osiński, Sergey Levine, et al. Lm-nav: Robotic navigation with large pre-trained models of language, vision, and action. In *Conference on robot learning*, pages 492–504. PMLR, 2023.

[79] Dhruv Shah, Ajay Sridhar, Arjun Bhorkar, Noriaki Hirose, and Sergey Levine. Gnm: A general navigation model to drive any robot. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7226–7233. IEEE, 2023.

[80] Enxin Song, Wenhao Chai, Guanhong Wang, Yucheng Zhang, Haoyang Zhou, Feiyang Wu, Haozhe Chi, Xun Guo, Tian Ye, Yanting Zhang, et al. Moviechat: From dense token to sparse memory for long video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18221–18232, 2024.

[81] Ajay Sridhar, Dhruv Shah, Catherine Glossop, and Sergey Levine. Nomad: Goal masked diffusion policies for navigation and exploration. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 63–70. IEEE, 2024.

[82] Quan Sun, Yuxin Fang, Ledell Wu, Xinlong Wang, and Yue Cao. Eva-clip: Improved training techniques for clip at scale. *arXiv preprint arXiv:2303.15389*, 2023.

[83] Sinan Tan, Mengmeng Ge, Di Guo, Huaping Liu, and Fuchun Sun. Knowledge-based embodied question answering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(10):11948–11960, 2023.

[84] Nguyen Van Toan, Minh Do Hoang, Phan Bui Khoi, and Soo-Yeong Yi. The human-following strategy for mobile robots in mixed environments. *Robotics and Autonomous Systems*, 160:104317, 2023.

[85] Hanqing Wang, Wei Liang, Luc V Gool, and Wenguan Wang. Towards versatile embodied navigation. *Advances in neural information processing systems*, 35:36858–36874, 2022.

[86] Hongcheng Wang, Andy Guan Hong Chen, Xiaoqi Li, Mingdong Wu, and Hao Dong. Find what you want: learning demand-conditioned object attribute space for demand-driven navigation. *Advances in Neural Information Processing Systems*, 36, 2024.

[87] Zihan Wang, Xiangyang Li, Jiahao Yang, Yeqi Liu, and Shuqiang Jiang. Gridmm: Grid memory map for vision-and-language navigation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15625–15636, 2023.

[88] Zihan Wang, Xiangyang Li, Jiahao Yang, Yeqi Liu, and Shuqiang Jiang. Sim-to-real transfer via 3d feature fields for vision-and-language navigation. *arXiv preprint arXiv:2406.09798*, 2024.

[89] 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12009–12020, 2023.

[90] 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. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6659–6668, 2019.

[91] Erik Wijmans, Abhishek Kadian, Ari Morcos, Stefan Lee, Irfan Essa, Devi Parikh, Manolis Savva, and Dhruv Batra. DD-PPO: Learning near-perfect pointgoal navigators from 2.5 billion frames. In *International Conference on Learning Representations (ICLR)*, 2020.

[92] Jimmy Wu, William Chong, Robert Holmberg, Aaditya Prasad, Yihuai Gao, Oussama Khatib, Shuran Song, Szymon Rusinkiewicz, and Jeannette Bohg. Tidybot++: An open-source holonomic mobile manipulator for robot learning. *arXiv preprint arXiv:2412.10447*, 2024.

[93] Qiaoyun Wu, Xiaoxi Gong, Kai Xu, Dinesh Manocha, Jingxuan Dong, and Jun Wang. Towards target-driven visual navigation in indoor scenes via generative imitation learning. *IEEE Robotics and Automation Letters*, 6 (1):175–182, 2020.

[94] Yuchen Wu, Pengcheng Zhang, Meiying Gu, Jin Zheng, and Xiao Bai. Embodied navigation with multi-modal information: A survey from tasks to methodology. *Information Fusion*, page 102532, 2024.

[95] Dejing Xu, Zhou Zhao, Jun Xiao, Fei Wu, Hanwang Zhang, Xiangnan He, and Yueting Zhuang. Video question answering via gradually refined attention over appearance and motion. In *ACM Multimedia*, 2017.

[96] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. Msr-vtt: A large video description dataset for bridging video and language. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5288–5296, 2016.

[97] Zifan Xu, Bo Liu, Xuesu Xiao, Anirudh Nair, and Peter Stone. Benchmarking reinforcement learning techniques for autonomous navigation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9224–9230. IEEE, 2023.

[98] Karmesh Yadav, Arjun Majumdar, Ram Ramrakhya, Naoki Yokoyama, Alexei Baevski, Zsolt Kira, Oleksandr Maksymets, and Dhruv Batra. Ovrl-v2: A simple state-of-art baseline for imagenav and objectnav. *arXiv preprint arXiv:2303.07798*, 2023.

[99] Karmesh Yadav, Ram Ramrakhya, Arjun Majumdar, Vincent-Pierre Berges, Sachit Kuhar, Dhruv Batra, Alexei Baevski, and Oleksandr Maksymets. Offline visual representation learning for embodied navigation. In *Workshop on Reincarnating Reinforcement Learning at ICLR 2023*, 2023.

[100] Naoki Yokoyama, Sehoon Ha, Dhruv Batra, Jiuguang Wang, and Bernadette Bucher. Vlfm: Vision-language frontier maps for zero-shot semantic navigation. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 42–48. IEEE, 2024.

[101] Naoki Yokoyama, Ram Ramrakhya, Abhishek Das, Dhruv Batra, and Sehoon Ha. Hm3d-ovon: A dataset and benchmark for open-vocabulary object goal navigation. *arXiv preprint arXiv:2409.14296*, 2024.

[102] Bangguo Yu, Hamidreza Kasaei, and Ming Cao. L3mvn: Leveraging large language models for visual target navigation. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3554–3560. IEEE, 2023.

[103] Licheng Yu, Xinlei Chen, Georgia Gkioxari, Mohit Bansal, Tamara L Berg, and Dhruv Batra. Multi-target embodied question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6309–6318, 2019.

[104] Zhou Yu, Dejing Xu, Jun Yu, Ting Yu, Zhou Zhao, Yueting Zhuang, and Dacheng Tao. Activitynet-qa: A dataset for understanding complex web videos via question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 9127–9134, 2019.

[105] Zhou Yu, Jun Yu, Yuhao Cui, Dacheng Tao, and Qi Tian. Deep modular co-attention networks for visual question answering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6281–

6290, 2019.

[106] Kuo-Hao Zeng, Zichen Zhang, Kiana Ehsani, Rose Hendrix, Jordi Salvador, Alvaro Herrasti, Ross Girshick, Aniruddha Kembhavi, and Luca Weihs. Poliformer: Scaling on-policy rl with transformers results in masterful navigators. *arXiv preprint arXiv:2406.20083*, 2024.

[107] Hang Zhang, Xin Li, and Lidong Bing. Video-llama: An instruction-tuned audio-visual language model for video understanding. *arXiv preprint arXiv:2306.02858*, 2023.

[108] Jiazhao Zhang, Chenyang Zhu, Lintao Zheng, and Kai Xu. Rosefusion: random optimization for online dense reconstruction under fast camera motion. *ACM Transactions on Graphics (TOG)*, 40(4):1–17, 2021.

[109] Jiazhao Zhang, Yijie Tang, He Wang, and Kai Xu. Asrodio: Active subspace random optimization based depth inertial odometry. *IEEE Transactions on Robotics*, 39 (2):1496–1508, 2022.

[110] Jiazhao Zhang, Nandiraju Gireesh, Jilong Wang, Xiaomeng Fang, Chaoyi Xu, Weiguang Chen, Liu Dai, and He Wang. Gamma: Graspability-aware mobile manipulation policy learning based on online grasping pose fusion. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1399–1405. IEEE, 2024.

[111] Jiazhao Zhang, Kunyu Wang, Rongtao Xu, Gengze Zhou, Yicong Hong, Xiaomeng Fang, Qi Wu, Zhizheng Zhang, and He Wang. Navid: Video-based vlm plans the next step for vision-and-language navigation. *Robotics: Science and Systems*, 2024.

[112] Yue Zhang, Ziqiao Ma, Jialu Li, Yanyuan Qiao, Zun Wang, Joyce Chai, Qi Wu, Mohit Bansal, and Parisa Kordjamshidi. Vision-and-language navigation today and tomorrow: A survey in the era of foundation models. *ArXiv*, abs/2407.07035, 2024. URL https://api.semanticscholar.org/CorpusID:271064503.

[113] Zhen Zhang, Jiaqing Yan, Xin Kong, Guangyao Zhai, and Yong Liu. Efficient motion planning based on kinodynamic model for quadruped robots following persons in confined spaces. *IEEE/ASME Transactions on Mechatronics*, 26(4):1997–2006, 2021.

[114] Duo Zheng, Shijia Huang, Lin Zhao, Yiwu Zhong, and Liwei Wang. Towards learning a generalist model for embodied navigation. *arXiv preprint arXiv:2312.02010*, 2023.

[115] Lintao Zheng, Chenyang Zhu, Jiazhao Zhang, Hang Zhao, Hui Huang, Matthias Niessner, and Kai Xu. Active scene understanding via online semantic reconstruction. In *Computer Graphics Forum*, volume 38, pages 103–114. Wiley Online Library, 2019.

[116] Fangwei Zhong, Peng Sun, Wenhan Luo, Tingyun Yan, and Yizhou Wang. Ad-vat+: An asymmetric dueling mechanism for learning and understanding visual active tracking. *IEEE transactions on pattern analysis and machine intelligence*, 43(5):1467–1482, 2019.

[117] Fangwei Zhong, Peng Sun, Wenhan Luo, Tingyun Yan, and Yizhou Wang. Towards distraction-robust active visual tracking. In *International Conference on Machine Learning*, pages 12782–12792. PMLR, 2021.

[118] Fangwei Zhong, Xiao Bi, Yudi Zhang, Wei Zhang, and Yizhou Wang. Rspt: reconstruct surroundings and predict trajectory for generalizable active object tracking. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 3705–3714, 2023.

[119] Fangwei Zhong, Kui Wu, Hai Ci, Churan Wang, and Hao Chen. Empowering embodied visual tracking with visual foundation models and offline rl. In *European Conference on Computer Vision*, pages 139–155. Springer, 2024.

[120] Gengze Zhou, Yicong Hong, and Qi Wu. Navgpt: Explicit reasoning in vision-and-language navigation with large language models. *arXiv preprint arXiv:2305.16986*, 2023.

[121] 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.

[122] Deyao Zhu, Jun Chen, Kilichbek Haydarov, Xiaoqian Shen, Wenxuan Zhang, and Mohamed Elhoseiny. Chatgpt asks, blip-2 answers: Automatic questioning towards enriched visual descriptions, 2023.

[123] Fengda Zhu, Yi Zhu, Vincent Lee, Xiaodan Liang, and Xiaojun Chang. Deep learning for embodied vision navigation: A survey. *arXiv preprint arXiv:2108.04097*, 2021.

[124] Yuke Zhu, Roozbeh Mottaghi, Eric Kolve, Joseph J Lim, Abhinav Gupta, Li Fei-Fei, and Ali Farhadi. Target-driven visual navigation in indoor scenes using deep reinforcement learning. In *2017 IEEE international conference on robotics and automation (ICRA)*, pages 3357–3364. IEEE, 2017.

[125] Weiqin Zu, Wenbin Song, Ruiqing Chen, Ze Guo, Fanglei Sun, Zheng Tian, Wei Pan, and Jun Wang. Language and sketching: An llm-driven interactive multimodal multitask robot navigation framework. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1019–1025. IEEE, 2024.

CONTENTS

IX. TASK DEFINITION

We introduce the details of four embodied navigation tasks that are included in our paper.

A. *Vision-and-language Navigation (VLN)*

Vision-and-language navigation [44] requires the agent to follow the instruction by moving between given landmarks and stopping at the described destination within unseen environments. The instruction of VLN is at free-form which describes a trajectory of landmarks and the motions between these landmarks. The landmarks and motions are open-vocabulary. VLN is widely regarded as a challenging task because it has to understand the free-form instruction, align the navigation history with instruction, and do path planning. Despite the fact that many works consider using pre-build landmark graphs [120, 61, 121] to simplify VLN, we consider a more practical setting that uses a continuous environment (VLN-CE [44]). Currently, there are two mainstream VLN-CE datasets: VLN-CE R2R [44] and VLN-CE RxR [47]. We provide examples of these instructions:

- **VLN-CE R2R**: Walk down the hallway along the banister railing on the upper floor of the home. Walk through the open door next to the staircase. Walk into the room, which has a couch and chairs around a coffee table.
- **VLN-CE RxR**: You are starting in the corner of a living room. Turn around to find a clock hanging on the wall in the hallway. Take two steps toward it. Turn right and walk straight, passing between the blue couch and the kitchen. You should now be looking through windows into the backyard. To your right is an open patio, and to your left are four framed black-and-white pictures. You've completed the instructions.

In the benchmark VLN-CE R2R and VLN-CE RxR, the agent is required to navigate each of the landmarks in the given order of the instruction and stop within 3 meters of the destination. The max navigation step is 500 steps.

B. *Object Goal Navigation (ObjectNav)*

In ObjectNav [76], an agent is initialized at a random starting position and orientation in an unseen environment and asked to find an instance of an object category ('find a chair') by navigating to it. The agent should explore the environments to track the location location that could locate target objects, and then identify the target object and stop nearby. Specifically, we follow the Habitat-matterport 3D dataset, which requires the agent to search a specific category object, which could be one of a category set including couch, bed, chair, toilet, plant, and TV. In the benchmark HM3D, the trajectory is considered a success if the target stops within 1 meter of the target object under 500 steps.

C. *Embodied Question Answering (EQA)*

Embodied Question Answering [21] is a complicated task, which involves answering questions by interacting with and navigating within an unseen 3D environment. Following the setting of MP3D-EQA, the agent is first given a question related to an object, like the color or location, and then the agent is required to identify the described target and return a natural language that directly answers the question. Here are some examples of the questions:
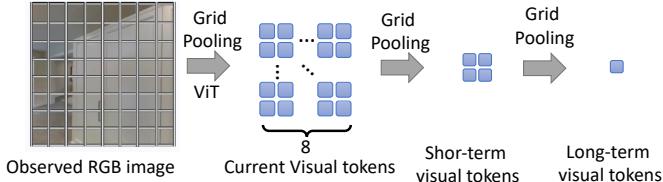
Fig. 9: **Grid pooling**. We add a visualization of grid pooling across different types of observations.

- What room is the chair located in?
- What color is the bed?
- What color is the couch in the living room?

In the benchmark MP3D-EQA, the agent is required to find the target and answer the question within 500 steps. And the episode will be considered a success if the answer matches the GT answer.

D. Human Following

The three tasks discussed above primarily address static and fixed scenarios, with limited applicability to dynamic environments. In contrast, human following [28, 37, 125], as a classic robot target tracking problem, focuses on enabling robots to interact effectively in dynamic settings, making it particularly well-suited for real-world applications.

Traditional human following tasks often rely on modular-based approaches [33], mainly focusing on the robot's following strategies [84] and controller design [41]. These studies typically assume that the target to be tracked is either known or specified at the beginning, an assumption that fails to account for the complexities of real-world environments. In this paper, we introduce a novel human-following task driven by natural language instruction, where the robot should locate and follow a specific human that aligns with the given description in a potentially crowded and dynamic environment.

We developed a benchmark for this task based on Habitat 3.0 [68]. In each episode, multiple humans with diverse characteristics are randomly placed in the scene, and the robot is initialized near the target human to be followed, ensuring that this human is within the robot's initial observation. The robot interprets natural language instructions, such as *"follow the man wearing a blue shirt and black pants"* or *"stay behind the woman in yellow"*, to locate the individual that best matches the description. It then tracks and follows the person's movements, dynamically adapting until reaching the pre-defined navigation goal. An episode is deemed successful if the robot stops within 2 meters of the correct human and faces him/her at the end.

X. IMPLEMENTATION DETAILS

In this section, we provide more implementation details of Uni-NaVid.

A. Online Visual Token Merging

Online visual token merging is the key technique of Uni-NaVid to enable efficient encoding of a long horizon of ego-centric videos. Specifically, we organize the online captured

frames. When the model only receives one frame, that frame becomes the current observation. Then we extract the visual tokens and leverage grid pooling (See Fig. 9). We split the image into 8×8 and ach grid is conducted an average operation, leading to the final 64 tokens for current observations. Then with more incoming frames, we perform grid pooling to older current observation tokens, which leads to 2×2 tokens, and append them into a short-term memory buffer.

If the time step is over 65 steps, then the oldest short-term frame will be pooped out and then is performed grid pooling to 1 token. This token is a long-term visual token which then will be inserted into the long-term visual token list if the long-term visual token list is empty or the cos similarity is smaller than τ. Here, we use $\tau = 0.95$, which is obtained empirically [80], which achieves a balance between the efficiency and effective.

B. Token Organization

To facilitate the understanding of the large language model, we have to organize the tokens, including observation tokens, instruction tokens, and special tokens. Specifically, we use observation indicator tokens to indicate the following parts are visual tokens. Besides, we add an image separator token between the adjacent visual tokens of each frame (following existing VLMs [30, 57]), this is crucial to distinguish the visual information inherent from different frames. Finally, if the task is navigation, we add a navigation special token <NAV> to indicate the task is navigation. Once the model understands the navigation special token, it will directly output action tokens.

Note that, it is important to use a navigation special token which could address the ambiguity problem under the embodied question-answering task because a large language model could confused about whether to directly answer the question or output actions to the target. The supported experiments can be found in the main paper Table 9.

C. Traing Strategy

We follow the training strategy of NaVid [111] in a two-stage manner. In the first stage, we firstly pre-train the projector of Uni-NaVid with the Image QA dataset then finetune both the projector and LLM with the Video QA dataset. We collect the data from LLama-Vid [51] and Pandm [19]. In the second state, we train both projector and LLM with collected navigation data. The default parameters are borrowed from NaVid [111].

XI. DATA PREPARATION DETAILS

To train Uni-NaVid, we required massive navigation data across different navigation tasks. It is extremely challenging to collect a large number of high-quality annotated data in the real world. Therefore, we collect navigation data in the synthetic environments and we describe the details in the following sections.

A. Data Collection

Navigation samples. We define the navigation samples including a navigation history video, corresponding instructions, and future actions (we use four actions). Here the navigation

history video is accumulated frames to time step T, which can be indicated as a set of frames $\{\mathbf{x}_t\}_{1:T}$

Vision-and-language navigation. We collect VLN navigation samples in the VLN-CE simulator [44, 75]. We use the training split of R2R dataset [44] and RxR dataset [46], which include the ground-truth navigation trajectory and corresponding instructions within the HM3D datasets [70]. Therefore, we deploy the agent to follow the GT trajectory and render RGB images during navigation. In this case, we collect 0.64 M navigation video-action samples.

Besides GT navigation samples, we also use DAGGER [74] to collect more diverse navigation samples, nearly 1.69 M navigation samples. Specifically, we run Uni-NaVidin the training split and collect the expert actions by using a deterministic path planning method [77] to the next non-arrived landmark.

We also collect a sub-split ($70k$) of previously collected VLN data (randomly sample the trajectories which are smaller than 20 steps) and augment the instruction with low-level actions, *e.g., "move forward 4 steps, then turn right 3 steps.".* We find that Uni-NaVidcan easily master the low-level instructions, but the performance drops when the low-level instruction expands significantly.

Object goal navigation. We collect the object goal navigation data in the HM3D datasets [70] in Habitat simulator [75]. Here, the agent is initialized in unseen environments and is required to find one object that could be a category set of chair, couch, TV, toilet, bed, and plant. We deploy L3MVN [102] in the HM3D training split and collect the navigation trajectory. We only collect 483 k navigation trajectory which is successful in finding the target. And we use the instruction template: "*Search for a/an* [Object]." Note that, using the shortest path to the target object as the navigation video samples leads to an extremely low performance (30.1% SR) because the model can not learn to explore or recover from mistakes.

Embodied question answering. We collect 240 k video-action samples from MP3D-EQA dataset [21] and 10 k video-answering samples. The video-action samples are rendered based on the GT trajectory of MP3D-EQA, and use the question as the instruction. And the video-answering samples are rendered frames of full trajectory, and we use the GT answering as the answer of large language model of MP3D-EQA.

Human following. We collect 544 k video-action samples from a self-build human following environment based on the Habitat 3.0 simulator. Specifically, we generate a large amount of human-following data based on the HM3D dataset [70]. For each episode, we first determine the total number of humans (2–6) based on the area of the scene, assign an ID to the target human to be followed, and specify the language instruction. Next, we randomly initialize the starting positions of the humans in the scene and place the robot near the target human, ensuring that the target is visible in the robot's camera view. Finally, we randomly generate multiple navigation waypoints for each human and use a low-level path planner to guide them to these waypoints sequentially. During this process, the robot
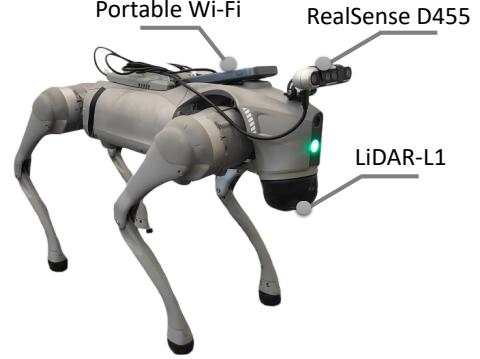

Fig. 10: **Robot setup.** We use Unitree GO2 as our embodiment, and we mount RealSense D455, a portable Wi-Fi and a LiDAR-L1. Note that, our model only takes RGB frames as input. The portable Wi-Fi is used for communication with the remote server and the Lidar is used for the local controller API of Unitree Dog.

continuously receives the target human's position in each step and uses a local planner to follow the human while avoiding obstacles nearby until the human reaches the final waypoint.

B. Instruction Augmentation

We collect instructions from various datasets, and the quality of the instructions is limited in diversity. Especially for object goal navigation, the fixed template could cause the instruction to become a task indicator, which could damage the performance. In this case, we use ChatGPT, to augment the instructions. The prompts are listed as follows:

> Given a robot navigation task instruction, rephrase the instruction to make its grammar and structure more diverse while preserving its original meaning. Additionally, ensure all grammatical errors in the instruction are corrected. Use varied sentence structures and descriptions for objects and directions to enhance linguistic diversity. Keep the instructions clear and concise to ensure they remain suitable for a robot navigation system.

We find the instruction augmentation could increase the performance of VLN ($+2.31\%$ in SR) and ObjectNav($+3.7\%$ in SR). We believe the cross-dataset instruction augmentation could be a promising topic to further investigate.

XII. REAL-WORLD DEPLOYMENT

A. Robot Setup.

We provide a visualization of our robotic dog in Fig. 10. Our robotic dog is Unitree GO2 and we mount a RealSense D455 camera on the head of the robotic dog. Here, we only use the RGB frames with a resolution of 640×480 in the setting of $90°$ HFOV. We also mount a portable Wi-Fi at the back of the robot dog, which is used to communicate with the
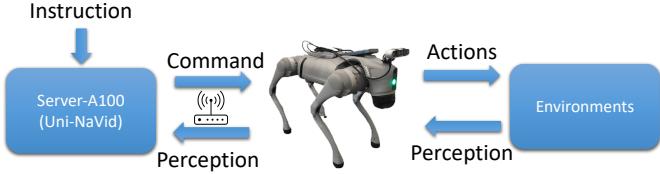
Fig. 11: **Real-world system architecture.** We deploy our model at a remote server, and the robot communicates with the server through the Internet.

remote server (send captured images and receive commands). Unitree GO2 is integrated with a LiDAR-L1, which is only used for local motion planning.

Note that Uni-NaVid does not rely on any odometry algorithms [108, 109] or noiseless depth [76, 115, 48], making it easy to deploy in real-world environments.

B. Real-world System Architecture

We provide a visualization of the real-world setup in Fig. 11. Our model is deployed on a remote server equipped with an NVIDIA A100 GPU. During navigation, the server receives navigation instructions and images captured by the robotic dog through the Internet. To ensure efficient communication, the images are compressed before transmission. After processing the newly captured images, the model generates action commands (FORWARD, LEFT, RIGHT, and STOP) and sends them to the robotic dog. Upon receiving these commands, the robotic dog executes the actions using a local motion planning model (specifically, the off-the-shelf model provided by Unitree Dog). Leveraging our online token merging strategy, the model processes newly captured images efficiently, requiring approximately 0.2 seconds for inference, while Internet communication (image transmission and command reception) takes about 0.3 seconds.

Non-blocking navigation. After receiving a set of four action commands, the robotic dog executes them sequentially. Upon completing each action, the robot captures a new image and transmits it to the server. In cases where multiple action commands are generated for a single step, the robot prioritizes and executes the most recent command, as it reflects the latest planning outcome.

XIII. EXPERIMENTS

A. Experiment Setup

1) Benchmark.: We conduct extensive experiments on public benchmarks across different tasks. We celebrate them as follows: *Vision-and-language navigation.* We conduct experiments on the Val-Unseen split of VLN-CE R2R (Main paper Table 2) and RxR (Main paper Table 3). They contain novel environments and novel instructions. *Embodied Question answering.* We conduct experiments on the validation set of the MP3D-EQA [90], where the questions and scenes are novel. Besides, we also conduct experiments (Table XIII) on OpenEQA [63], which is a benchmark to evaluate the

performance of understanding the marionettes with egocentric video. *Human following.* In Table 5 of the main paper, the experiment is conducted on a self-build benchmark based on the HM3D dataset, which also served as the source of the training dataset. Additionally, we also build two benchmarks based on the HSSD dataset and MP3D dataset, respectively, and evaluate the human following capability of Uni-NaVid across different datasets and scenes (Table XIV).

2) Metrics.: We use the default metrics of each benchmark. For vision-and-language navigation and object goal navigation, [44, 76, 94], we use SR, OSR, and SPL as metrics. Specifically, SR (Success Rate) measures the proportion of tasks in which the agent successfully reaches the target location (in VLN the distance threshold is 3 m, and in ObjectNav is 1 m) within the allowed time steps (Up to 500 steps). OSR (Oracle Success Rate) extends SR by accounting for the agent's proximity to the goal, considering the task successful if the agent is close enough, even without directly stopping at the target. SPL (Success weighted by Path Length) evaluates the agent's efficiency by combining success with the optimality of its trajectory, penalizing longer-than-necessary paths while rewarding those closer to the shortest possible route.

For human following navigation, we use SR, FR, and CR metrics. SR (Success Rate) measures the proportion of events in which the agent successfully follows the target human to the endpoint. FR (Following Rate) refers to the proportion of steps in which the agent successfully follows the target human for each step. CR (Collision Rate) refers to the proportion of episodes in which the agent collides with the human during the movement.

For embodied question answering, we use ACC metric, which directly measures the percentage of correct answers. For ScnaQA [7], we use EM, BLUE, ROUGE, METEOR, CIDEr as metrics. Specifically, EM (Exact Match) evaluates the percentage of predictions that exactly match the reference answers. BLEU (Bilingual Evaluation Understudy) measures the n-gram overlap between the generated text and the reference, rewarding precise matches in shorter sequences. ROUGE (Recall-Oriented Understudy for Gisting Evaluation) assesses the overlap of word sequences, focusing on recall and capturing the informativeness of the generated text. METEOR (Metric for Evaluation of Translation with Explicit ORdering) combines precision and recall, using synonym matching and stemming to account for variations in phrasing. Lastly, CIDEr (Consensus-based Image Description Evaluation) measures the similarity of generated responses to multiple references by capturing human consensus, emphasizing relevance and diversity in the output.

For video-question answering benchmark MSVD-QA [95], MSRVTT-QA [95], and ActivityNet-QA [104], we use ACC and Score metrics. Specifically, ACC (Accuracy) measures the proportion of correctly answered questions, providing a straightforward assessment of the model's overall correctness. Score, on the other hand, evaluates the quality of the generated answers using GPT (GPT 3.5 in implementation) as the zero-shot evaluation assistor to assign relative scores on a scale of 1 to 5 for generated answers.

Method	Simple Ins.	Complex Ins.
NaVid [111]	80%	20%
Uni-NaVid	92%	84%

TABLE XI: **Real-world VLN experiments.** We compare our method with NaVid on two types of instructions: simple instructions (25) and complex instructions (25).

Method	Observation			VLN-CE RxR Val-Unseen				
	Odom.	Depth	S.RGB	TL	NE↓	OS↑	SR↑	SPL↑
LAW [73]	✓	✓	✓	4.01	10.87	21.0	8.0	8.0
CM2 [26]	✓	✓	✓	12.29	8.98	25.3	14.4	9.2
WS-MGMap [16]	✓	✓	✓	10.80	9.83	29.8	15.0	12.1
ETPNav.FF [88]	✓	✓	✓	-	8.79	36.7	25.5	18.1
Seq2Seq [44]		✓	✓	1.16	11.8	5.02	3.51	3.43
CMA [44]		✓	✓	5.09	11.7	10.7	4.41	2.47
A^2Nav† [17]		✓		–	–	–	16.8	6.3
NaVid [111]		✓		10.59	8.41	34.5	23.8	21.2
Uni-NaVid		✓		8.3	**8.08**	**40.9**	**29.5**	**28.1**

TABLE XII: **Vision-and-language navigation (RxR).** Comparison on VLN-CE RxR [47] Val-Unseen. †: indicates zero-shot methods.

Method	EM-EQA		
	ScanNet	HM3D	ALL
	Eq. (1)	Eq. (1)	Eq. (1)
Blind LLMs			
GPT-4	32.5	35.5	33.5
LLaMA-2	27.9	29.0	28.3
Socratic LLMs w/ Frame Captions			
GPT-4 w/ LLaVA-1.5	45.4	40.0	43.6
LLaMA-2 w/ LLaVA-1.5	39.6	31.1	36.8
Socratic LLMs w/ Scene-Graph Captions			
GPT-4 w/ CG	37.8	34.0	36.5
LLaMA-2 w/ CG	31.0	24.2	28.7
GPT-4 w/ SVM	40.9	35.0	38.9
LLaMA-2 w/ SVM	36.0	30.9	34.3
Multi-Frame VLMs			
GPT-4V	51.3	46.6	49.6
Human Agent	87.7	85.1	86.8
Uni-NaVid	41.4	38.1	40.3

TABLE XIII: **Embodied video question answering.** Comparison on OpenEQA benchmark. EM-EQA results are broken down by data source (ScanNet, HM3D, and ALL). GPT-4V scores are calculated on a subset of 500 OpenEQA questions due to API limitations.

B. Real-world Experiments

We conduct real-world experiments to study the generalizability of our method in the real world. Specifically, we leverage the VLN task, which includes both landmarks and motions, to evaluate our method with the previous VLN method NaVid [111]. Following previous work [111], we designed two types of instructions for different difficulties (25 simple instructions and 25 complex instructions). The simple instructions, which require the agent to navigate to a single robot landmark and stop; (2) complex instructions, which require the agent to follow a series of simple instructions.

We list some examples of used instructions in experiments. Simple instruction examples:

- Move forward 1 step, then stop.
- Turn left 5 steps, then stop.
- Move to the right chair, then turn left.

Complex instructions:

- Move forward 5 steps, then turn left 4 steps, and finally turn right 5 steps.
- Go straight to the chair, then turn right and move to the door, stop by the door.
- Turn a large right, and go forward to a plant, then right at the plant and move to the TV, stop close to TV.

Here we present our results in Table XI. We find that both Navid and our method can achieve high SR in simple instructions, However, for complex instructions, our method shows significant improvements. This demonstrates the superiority of our method over existing methods. We add more visual results to the attached video.

C. More Benchmark Experiments

Cross-dataset Vision-and-Language Navigation Evaluation. We evaluate the cross-dataset performance of our method on VLN-CE RxR by excluding the VLN-CE RxR data at training time. The results are presented in Table XII. Notably, even without training on RxR, our method still outperforms existing approaches. However, a significant decrease in all metrics is observed compared to when RxR training data is included. We hypothesize that this discrepancy arises from differences in trajectory characteristics: trajectories in the R2R dataset have relatively uniform lengths (approximately 10 meters), whereas those in RxR exhibit greater diversity, ranging from approximately 2 meters to 20 meters. This disparity constrains our method's performance on the RxR dataset and underscores the importance of training on diverse trajectories.

OpenEQA benchmark [63]. To further evaluate our method on embodied question answering, we conduct experiments in OpenEQA, which require the methods to answer questions by analyzing an egocentric video. The results are shown in Tab. XIII. We find our method achieves competitive performance to the strongest commercial used Vision-Language model such as GPT-4V. Besides, by directly observing and understanding the video sequences, our method does not need additional frame caption or Socratic-style promoted [63] used in other LLMs.

Cross-Environments Following. We evaluate our method in novel environments, including HSSD [42] (synthetic environments) and MP3D [11] (reconstructed environments). The results, presented in Table XIV, demonstrate that our approach consistently outperforms baseline methods in both SR and FR metrics across HSSD and MP3D. Notably, our method achieves significant improvements in HSSD, likely due to the absence of reconstruction artifacts in synthetic environments, which reduces the likelihood of the robot being stuck. For the CR (collision rate) metric, IBVS adopts a highly conservative strategy that maintains a considerable distance from the target (i.e., a large bounding box). While this results in a lower CR,

Method	HF-HSSD			HF-MP3D		
	SR↑	FR↑	CR↓	SR↑	FR↑	CR↓
PoliFormer [106]	2.67	20.81	0.97	2.62	16.59	1.43
PoliFormer* [106]	26.97	54.20	10.01	25.42	47.80	8.72
PoliFormer† [106]	27.10	53.96	9.34	21.90	41.42	7.15
IBVS* [28]	66.32	80.26	0.19	56.86	68.91	1.33
IBVS† [28]	65.36	80.33	**0.15**	58.15	65.83	**0.77**
Uni-NaVid	**81.65**	**89.34**	1.33	**69.80**	**78.96**	2.99

TABLE XIV: **Human following.** Comparison on Human Following HSSD Benchmark (HF-HSSD) and Human Following MP3D Benchmark (HF-MP3D). *: Methods use Grounding-DINO [58] as the open-vocabulary human detector. †: Methods use the ground-truth bounding box provided by the simulator.

Design	VLN (SR↑)	ObjNav (SR↑)	EQA (ACC↑)	Follow (SR↑)
4 tokens for Curr. Obs.	43.4	50.2	40.5	-
1 token for Curr. Obs.	35.1	45.3	31.2	-
1 token for Short. Obs.	46.2	69.2	44.9	60.3
τ=0.9	40.2	70.0	43.2	57.8
τ=0.95	48.7	73.7	47.3	61.2
τ=0.99	49.6	70.2	48.1	57.2

TABLE XV: **Ablation study** of token number and τ

it adversely affects FR and SR performance.

D. More Ablation study

We conduct additional ablation studies to validate the effectiveness of our key design components. The results are presented in Table XV. Notably, we observed a significant performance drop when the number of current observation tokens was reduced. In particular, the following task failed across all sequences, potentially due to the inability of a smaller number of tokens to provide sufficient information for tracking human motion. Furthermore, we found that increasing the value of τ led to improved performance. This outcome is intuitive, as a higher τ retains more long-term observation tokens, which are crucial for understanding navigation history.

E. Time Analyze

To evaluate the efficiency of our method, we present the average running time and token count across different time steps in Fig. 12. Our approach is compared against existing models that also employ token merging strategies. The results
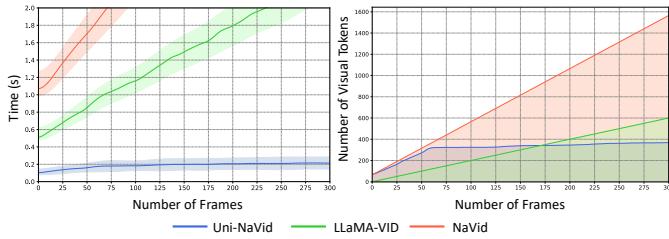

Fig. 12: **Time efficiency visualization.** We provide the average running time and the number of tokens across different time steps.
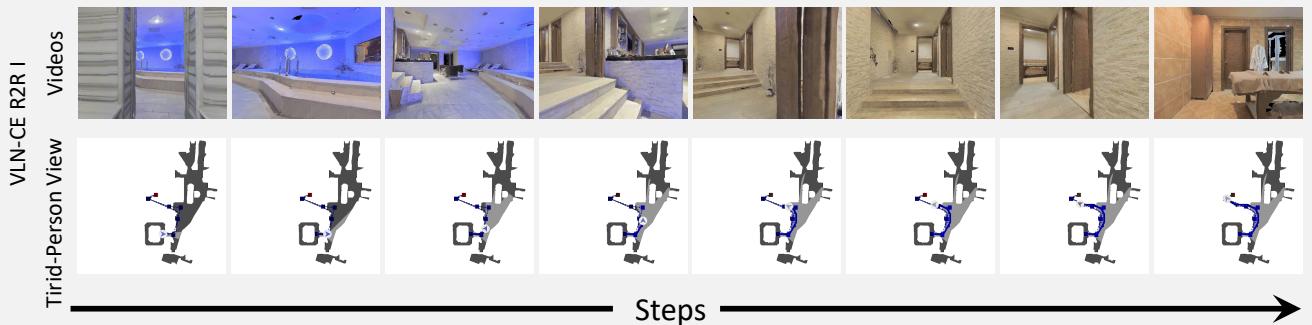
demonstrate that our method is more efficient which achieves an inference time of approximately 0.2 seconds. Moreover, our method maintains consistent running times, whereas the running times of existing methods increase significantly. These advantages stem from two key factors: *First*, our model architecture is highly efficient, whereas other approaches, such as LLaMA-VID [51] and NaVid [111], rely on time-consuming operations like QFormer. *Second*, our online token merging strategy results in a gradual increase in token count, ensuring more stable inference times.
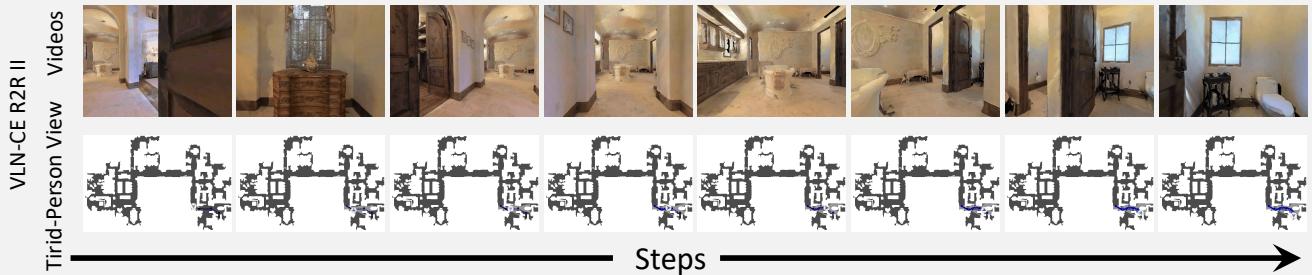
F. Qualitative Experiments

We provide visual results of our method on various benchmarks: Fig.13 illustrates results for VLN-CE R2R and RxR, Fig.14 for HM3D ObjectNav, and Fig. 15 for MP3D-EQA, Fig. 16 for HM3D human following. For additional visual results, please refer to the attached video.

Vision-and-language Navigation

Walk out of the sauna into the pool room and take a left. Walk towards the bar and take a left going up the stairs into the hallway. Go up the next part of the stairs and take at the first right into the massage room. Stop just inside the doorway on the right.

Turn left and walk forward towards the white tub. Turn right towards the first room that is next to the white tub, not on the corner. Walk inside the room. There should be a toilet bowl visible, and stop there.

You're standing in front of the frame of a door, turn to your left and you will notice an open door that leads to a bedroom, walk towards the foot of the bed in front of you in that room, and then turn left. You will see there's a mirror on your right side, and a picture hanging on the left wall. There's an open door in front of you, walk out of the room and head towards an open door on your left which leads to another bedroom. Once you're in the bedroom you will see a bed with blue sheets on your right, and an open door your left, walk past the open door on your left and as you Hook to your left in you will see a second open door that leads to a room. Walk into the bathroom and stand in front of the sink. That's your spot.

You are facing towards the glass window. Turn slightly right and exit the room. Then move forward, until you see an opening glass door. Enter into that room where it has a glass table in front of you and some wall hanging on the right, then take left and stop facing towards the blackboard, and this is your endpoint.
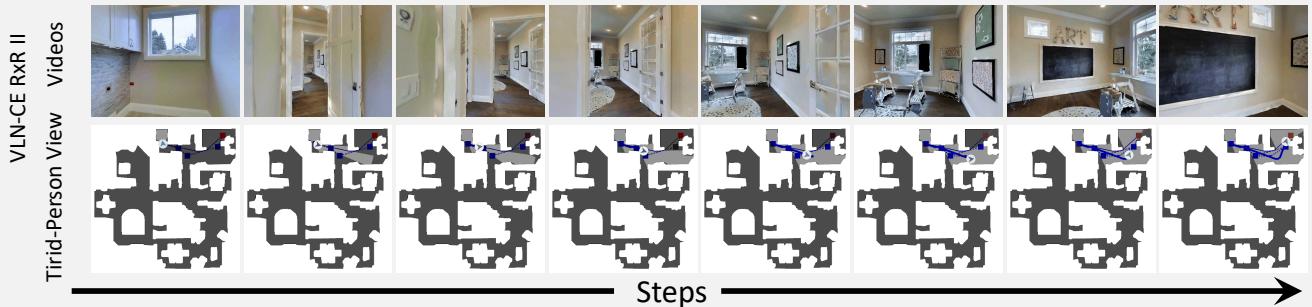

Fig. 13: Visual results of VLN on VLN-CE R2R and RxR.
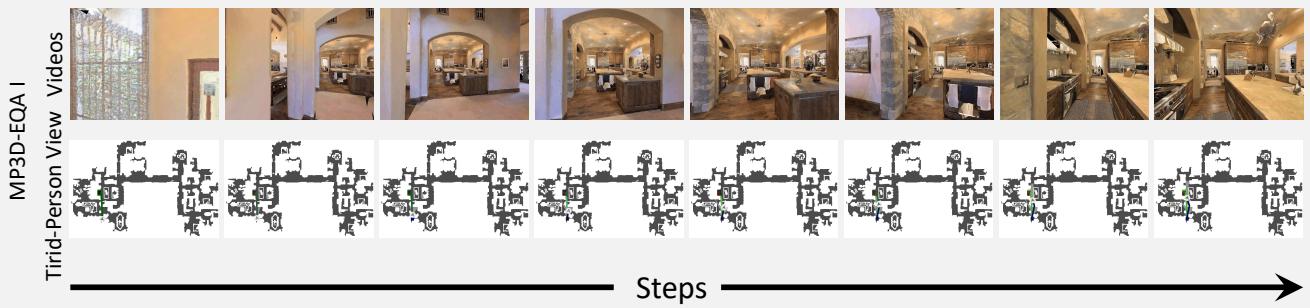
Object goal navigation

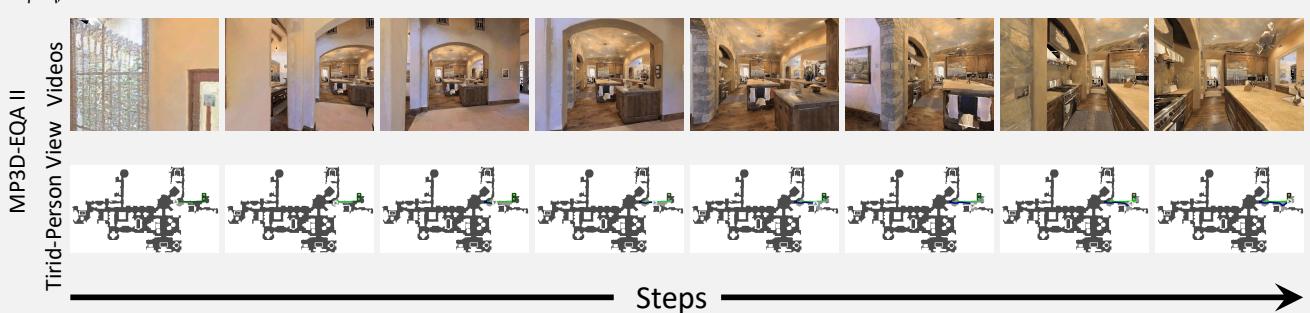Fig. 14: Visual results of HM3D ObjectNav.

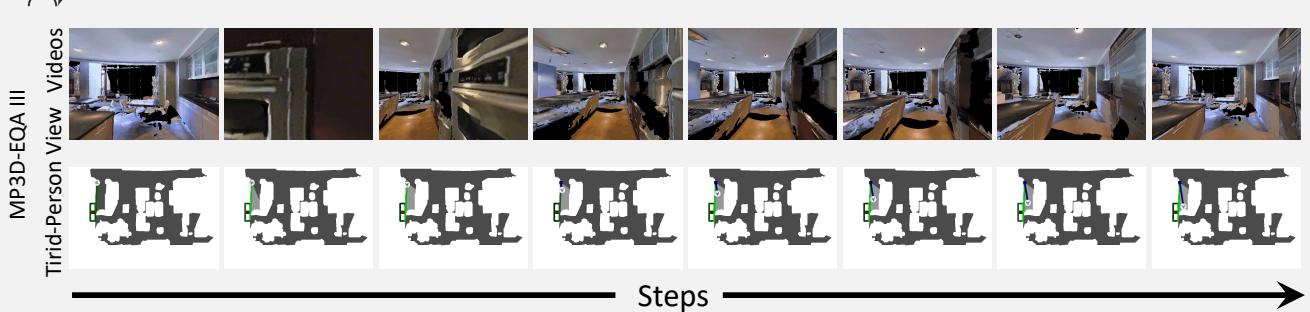# Embodied Question Answering

What color is the stove? **Uni-NaVid: Sliver**

What color is the wardrobe? **Uni-NaVid: Brown**

What room is the refrigerator located in? **Uni-NaVid: Kitchen**

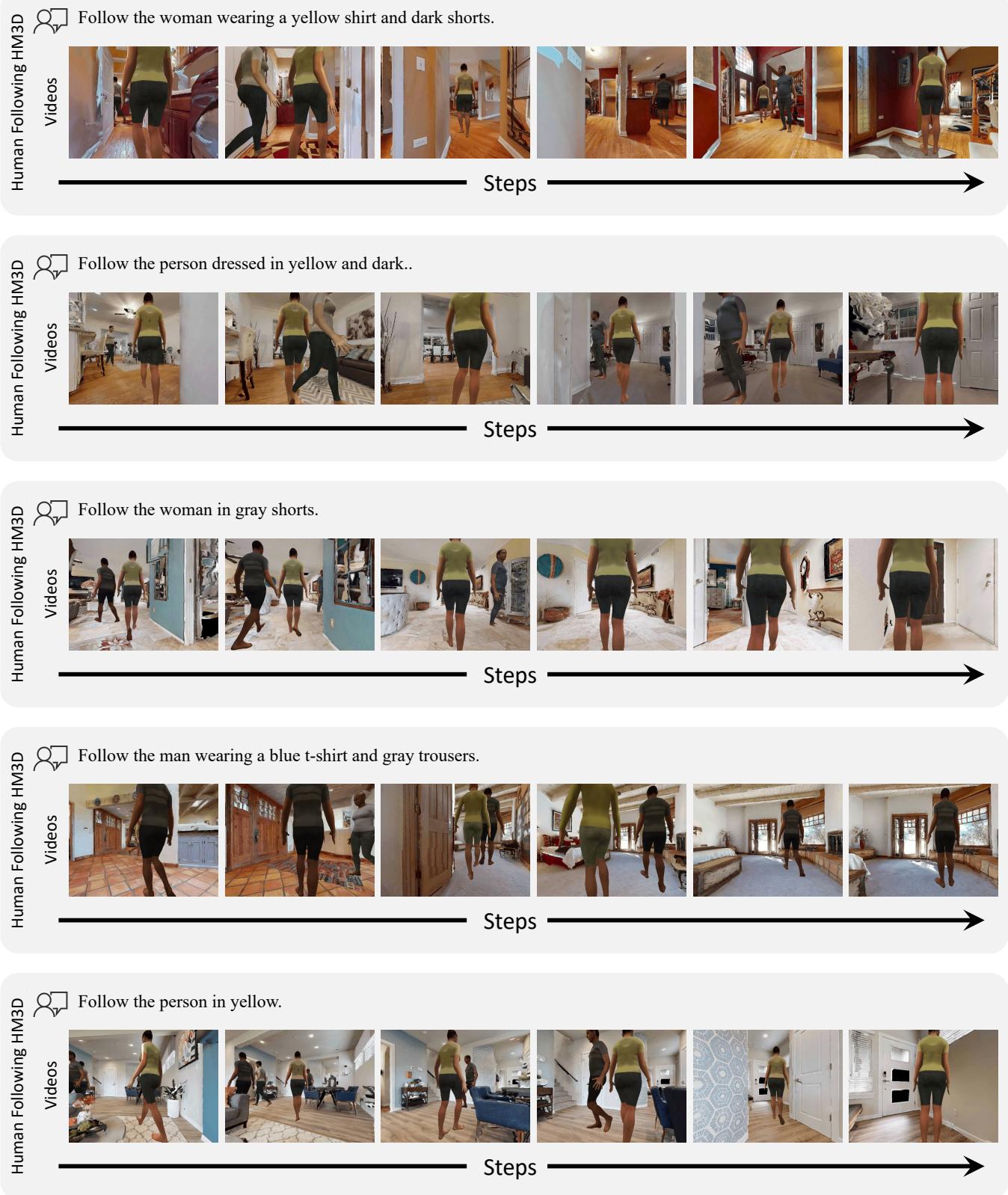What color is the sofa in the lounge? **Uni-NaVid: Brown**

Fig. 15: Visual results of MP3D-EQA.

Human Following

Fig. 16: Visual results of HM3D Human following.