

# Virtual Guidance as a Mid-level Representation for Navigation

Hsuan-Kung Yang<sup>1</sup>, Tsung-Chih Chiang<sup>1\*</sup>, Ting-Ru Liu<sup>1\*</sup>, Chun-Wei Huang<sup>1\*</sup>, Jou-Min Liu<sup>1\*</sup>, and Chun-Yi Lee<sup>1</sup>

**Abstract**—In the context of autonomous navigation, effectively conveying abstract navigational cues to agents in dynamic environments poses challenges, particularly when the navigation information is multimodal. To address this issue, the paper introduces a novel technique termed “Virtual Guidance,” which is designed to visually represent non-visual instructional signals. These visual cues, rendered as colored paths or spheres, are overlaid onto the agent’s camera view, serving as easily comprehensible navigational instructions. We evaluate our proposed method through experiments in both simulated and real-world settings. In the simulated environments, our virtual guidance outperforms baseline hybrid approaches in several metrics, including adherence to planned routes and obstacle avoidance. Furthermore, we extend the concept of virtual guidance to transform text-prompt-based instructions into a visually intuitive format for real-world experiments. Our results validate the adaptability of virtual guidance and its efficacy in enabling policy transfer from simulated scenarios to real-world ones.

## I. INTRODUCTION

Biological instincts are highly attuned to visual cues, a capability that is of paramount importance for executing navigation tasks effectively. For instance, Google Maps employs a combination of arrows and augmented reality to guide users to their desired destination. In contrast, navigational guidelines presented in textual or numerical formats, such as ‘turn 15 degrees at the next building’ or ‘proceed for 500 meters,’ are less straightforward and necessitate the high-level abstract information to be encoded into a form that can more easily be comprehended. Although certain studies have endeavored to transform these abstract navigational instructions into intermediate guidance signals (e.g., locational waypoints) that provide agents with spatial or directional cues [1]–[3], these often remain counterintuitive for model-free agents to easily grasp the environmental dynamics for effective navigation. This inadequacy becomes particularly pronounced when considering environments characterized by dynamic variables, such as moving objects, uncertainties, or other evolving conditions, where mere reliance on spatial or directional cues is inadequate. Moreover, the integration of disparate modalities (e.g., image observations and instructions) remains non-trivial due to the inherent challenges in reconciling the information derived from different modalities.

Despite the aforementioned challenges, there is still often a necessity for autonomous agents in visual navigation tasks to



Fig. 1. Demonstration of (a) the concept of virtual guidance, and (b) the agent’s behaviors in real-world scenarios in {🚦, 🌂, 🚗} and {🌂, 🌂} scenarios.

manage multiple modalities. These modalities are essential for both local control mechanisms and long-term planning strategies. Many previous navigation tasks have employed hierarchical frameworks to handle these diverse modalities. For example, the planning component may rely on non-visual cues such as LiDAR, GPS, or text instructions, while the local controller utilizes visual observations. Recent years have seen considerable success in integrating instructions into robot navigation tasks through such hierarchical methods [4]–[13]. However, such approaches necessitate that the agent first learn a mapping mechanism to correlate navigation information with observed images. In light of the extensive research on convolutional neural networks (CNNs) and their demonstrated efficacy in identifying complex interrelationships within an image, a potential promising avenue in the realm of visual navigation could lie in the direct representation of navigation instructions on visual perceptions. This could alleviate the cognitive burden on agents by rendering non-visual, abstract modalities more easily comprehensible.

Inspired by [14]–[18], we propose a method of representation for non-visual instructional or modal signals as visual format, which we term ‘virtual guidance.’ The virtual guidance signals, rendered as either colored paths or spheres, are superimposed on the semantic segmentation derived from the agent’s camera view, with the goal of guiding the agent toward a specific direction, as illustrated in Fig. 1 (a). This concept bears similarities to technologies such as Google Maps’ Line View, which incorporates Augmented Reality guidance based on Global Positioning Systems (GPS) or Visual Positioning Systems (VPS). To evaluate the effectiveness of the virtual guidance schemes, we developed a framework that comprises a planning module and a virtual guidance representation module. Our objective is to assess the efficacy of different kinds of vision-based virtual guidance schemes in comparison to the hybrid approach that concatenate segmentation with vector-based, non-visual guidance signals.

\* indicates equal contribution.

<sup>1</sup> Elsa Lab, Department of Computer Science, National Tsing Hua University, Hsinchu City, Taiwan.

This work has been submitted to the IEEE for possible publication. Copyright may be transferred without notice, after which this version may no longer be accessible.

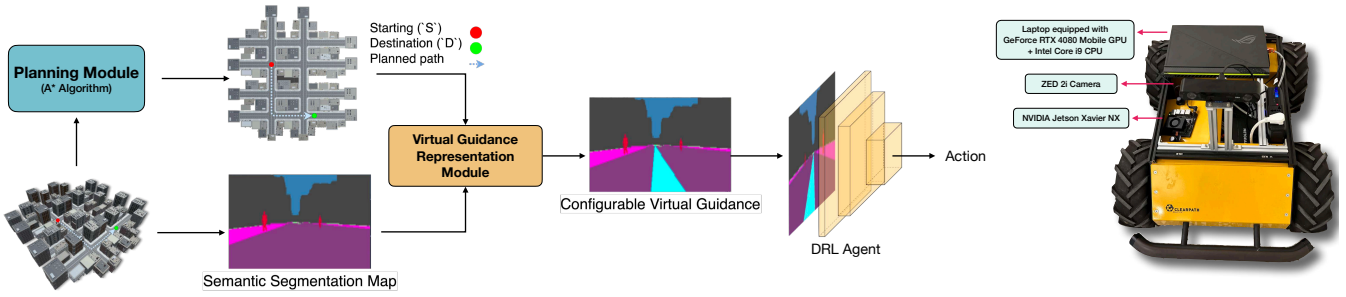


Fig. 2. **Left:** The overview of the simulation framework. **Right:** The hardware configuration used in our real-world experiments.

To validate our method, we provide simulated and real-world evaluations. For the former, we developed a configurable virtual city and multiple evaluation scenarios using Unity [19]. Such virtual environments enable the rendering of virtual guidance signals that are absent in existing off-the-shelf simulation platforms [20]. Our environments comprise dynamic objects moving at configurable speeds, and the agent is required to navigate while avoiding obstacles. We evaluated diverse experimental settings and observed that rendering guidance signals in the agent’s observations leads to superior performance in both adhering to designated routes and reaching the destination, compared to the hybrid guidance approach. Specifically, we compared the capabilities of different guidance schemes in *seen* and *unseen* conditions. Such examination offered empirical substantiation that that our proposed method excels in generalizability and obstacle avoidance proficiencies. Through impact, failure case, and trajectory analyses, we established that the agents trained under this virtual guidance paradigm outperformed those trained under the hybrid approach. For real-world evaluations, we extend the concept of virtual guidance to transform instructions from another modality into a more comprehensible form. Specifically, we employ text prompts to facilitate object detection through GroundingDINO [21], an open-vocabulary object detection model capable of zero-shot object detection. This process is illustrated in Fig. 1. Using the objects identified from the text prompts, we then generate virtual guidance and overlay it onto the agent’s observations. Our experiments validate the adaptability of the virtual guidance concept and the feasibility of transferring policies pre-trained in virtual scenarios to real-world ones.

## II. RELATED WORK

### A. Navigation and Guidance Mechanisms

Several navigation frameworks have been developed to direct an agent toward specified destinations. One common branch involves leveraging hierarchical frameworks to transmit navigational instructions or information to the agent [4]–[13]. Such non-vision-based guidance mechanisms can broadly be categorized as (1) implicit guidance and (2) vector-based guidance. The former offers the agent spatial knowledge through environmental cues, such as a localized map [13], [22], [23] or a set of waypoints [8], [24]. In contrast, the latter instructs the agent on specific actions or

orientations to adopt [12]. To elaborate on this latter category, previous research endeavors have experimented with giving agents directional guidance toward targets or waypoints using non-visual mechanisms [11], [12], [25], [26]. Beyond these two categories, recent literature has also delved into object-based navigation, where the agent is guided by images of the target object or area [27]. Some other research efforts represent waypoints as a sequence and instruct the agent to reach them in an ordered fashion [13], [22]–[26], [28]–[30], or through text-based instructions in vision-language frameworks. In this paper, the focus lies on explicit guidance realized through visual rendering of guidance signals. Unlike previous methodologies that concatenate observations with various modalities, our approach incorporates path or waypoint guidance directly into the agent’s observations.

### B. Mid-Level Representation based Navigation

To implement virtual guidance, this paper explores approaches for the representation of virtual guidance signals based on mid-level representations. Mid-level representations are abstract concepts that capture physical or semantic meanings, and are typically domain-invariant properties extracted from visual scenes. Such representations have found applications in robotics for conveying information from perception modules to control modules [18], [31]. These mid-level representations can assume various forms, such as depth maps, optical flow, and semantic segmentation, each possessing unique strengths and weaknesses in different scenarios [17]. A comprehensive, expressive, and interpretable mid-level representation is essential for the success of modular, learning-based frameworks. Existing research on navigation based on mid-level representations has primarily focused on facilitating obstacle avoidance or random path following for robotic agents, often in the absence of explicit instructional or guidance signals on direction or path [17], [18]. As a result, this study aims to extend this domain by introducing virtual guidance as a new form of mid-level representation.

## III. VIRTUAL GUIDANCE IN SIMULATED ENVIRONMENTS

### A. An Overview of the Simulation Framework

To investigate the feasibility of virtual guidance as a form of mid-level representation, we have developed a flexible framework using the Unity engine [19] and the Unity ML-Agents Toolkit [32]. This framework, illustrated in Fig. 2, is

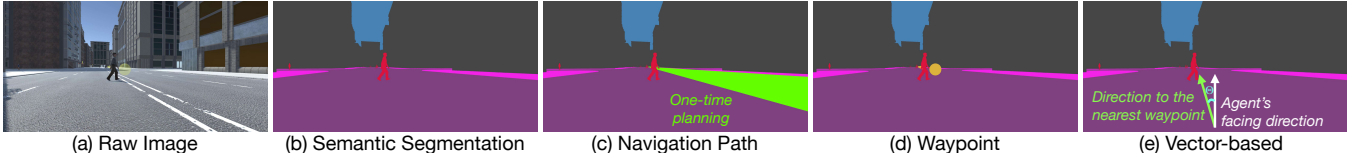


Fig. 3. The overview of different types of virtual guidance schemes compared to the vector based one.

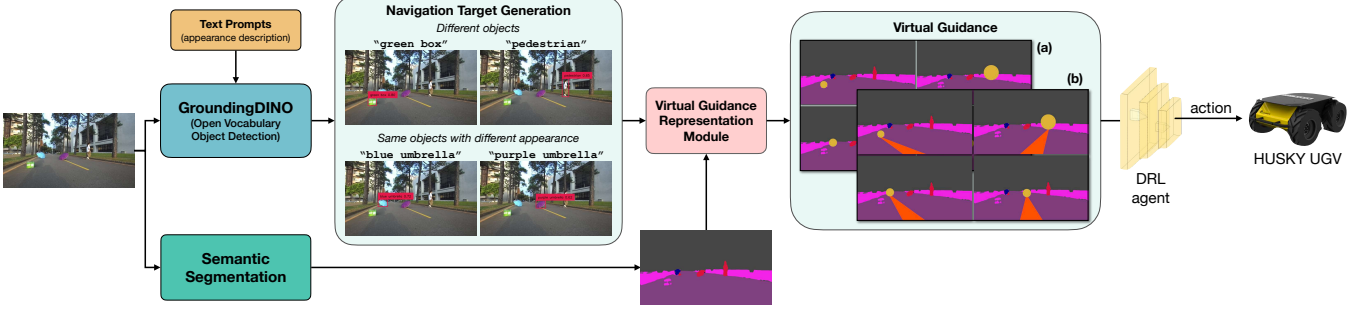


Fig. 4. The overview of the framework for deploying and validating the concept of virtual guidance in real-world scenarios.

designed to be fully configurable and enables the generation of guidance signals. The inputs of the agent are rendered in the form of semantic segmentation, an effective mid-level representation that can serve as an input observation for a Deep Reinforcement Learning (DRL) agent. The employment of semantic segmentation also facilitates transfer from virtual to real-world scenarios [18], [31]. The framework permits customization of various experimental settings, including different navigation paths and a range of training and evaluation environments featuring both static and dynamic objects. This design philosophy enables the exploration of diverse ways of presenting guidance signals to the agent, as well as various virtual guidance and vector-based approaches. The agent receives the guidance signals along with stacked semantic segmentation maps, and is tasked with processing these inputs to learn a policy to reach its intended destination.

### B. Virtual Guidance Generation Workflow

This section presents the workflow for generating virtual guidance, which consists of two primary components: (a) a *planning module* responsible for determining the navigation trajectory, and (b) a *virtual guidance representation module* tasked with rendering the virtual guidance for the DRL agent.

1) *Planning Module*: The function of the planning module is to facilitate the validation of the effectiveness of different guidance schemes. In the simulated environments, we assume that the planning module possesses awareness of the key locational parameters, i.e., the starting point  $S$ , the destination  $D$ , as well as the agent's position  $P_t$  at timestep  $t$ . This is to mitigate any potential errors in the localization process. It should be noted that in real-world scenarios, the planned trajectory can be generated in various ways, which is elaborated upon in the subsequent section. Under this assumption, a navigation trajectory can be planned between any two points, whether it be from  $S$  to  $D$  or  $P_t$  to  $D$ . The framework allows for configurability in the choice of planning algorithm, and the A\* algorithm [33] is employed

in the experiments of this study. The planned trajectory can either be rendered as a vision-based virtual guidance signal or transformed into vectors for the agent to interpret. The navigation trajectory can be generated in real-time between the agent's current position  $P_t$  and its destination  $D$ . While real-time planning offers advantages such as preventing the agent from losing track of its current position, it might not be feasible and practical under computational constraints [12]. In the simulated environments, we investigate two distinct configurations for the planning module to generate the navigation trajectory, including: (a) *real-time planning* between the DRL agent's current position  $P_t$  and the destination  $D$  at every timestep  $t$ , and (b) *one-time planning* from the starting point  $S$  to the destination  $D$  at the beginning of navigation.

2) *Virtual Guidance Representation Module*: Once the navigation trajectory is obtained from the planning module, virtual guidance can be generated through various approaches and rendered on semantic segmentation to enable the agent to recognize and interpret its semantic meaning. Different types of virtual guidance representations on semantic segmentation carry distinct visual meanings and may influence the agent's learning and evaluation behaviors in different ways. The proposed virtual guidance representation schemes are depicted in Fig. 3 and are elaborated as follows.

a) *Navigation Path*: In the first scheme, the navigation line obtained from the planning module is represented as a colored path on the semantic segmentation map. An example visualization is illustrated in Fig. 3 (c). Specifically, the navigation path is implemented as a 3D mesh in the simulated environments and projected onto the camera view plane. This rendered navigation path can be considered as a rich and informative signal that carries both semantic and guidance information. It highlights the permissible regions for the DRL agent and the route leading to the target location.

b) *Waypoint*: The second scheme generates a set of waypoints  $\mathcal{W}$  by segmenting the planned navigation trajec-



tory, where different waypoints are spaced with a regular distance from  $S$  to  $D$ . The waypoints serve as hints to instruct the agent to the destination. These waypoints are visualized as 3D virtual balls in the virtual environments and are projected onto the camera image plane. The visualization is presented in Fig. 3 (d). Unlike the first scheme, which utilizes a navigation path to provide dense and informative signals, the second one provides the waypoints as 3D virtual balls, which are sparse signals for the DRL agent to locate.

#### IV. VIRTUAL GUIDANCE IN REAL-WORLD SCENARIOS

In this section, we describe the methodology for representing virtual guidance in real-world settings, specifically through the use of text prompts. This demonstrates the flexible nature of our proposed approach, which can accommodate different modalities for generating virtual guidance.

##### A. An Overview of the Real-World Framework

To further validate the applicability of the proposed virtual guidance scheme in real-world scenarios, we have designed a specific task. The objectives of this task are twofold. First, the task seeks to verify the effective transferability of the pre-trained DRL agent's policy to real-world settings, where it follows the virtual guidance. Second, it aims to demonstrate that the virtual guidance scheme possesses the flexibility to adapt not only to trajectories generated by specific planning algorithms but also to instructions from diverse methods, provided these instructions can be translated into visual representations. This adaptability highlights the advantage of the virtual guidance scheme in eliminating the necessity for the agent to interpret inputs from multiple modalities. In the experiment, we provided a text prompt describing the appearance of the object to be navigated. We then assessed the agent's performance based on its ability to reach the correct target object as described in the text prompt. The framework for this experiment is illustrated in Fig. 4. Within this framework, we employed GroundingDINO [21], an open-vocabulary object detection model capable of zero-shot object detection, to generate detection results based on the given text prompt. The advantage of GroundingDINO, which employs a text encoder based on Bert-base [34], resides in its generalizability to detect a wide array of objects, a benefit derived from leveraging large-scale language models. The detection results are subsequently fed into the virtual guidance representation module described in Section III-B, where the objects detected according to the text prompt serve as guidance signals for the agent. These signals can be rendered in two different formats, similar to those used in simulated environments: (a) waypoints, or (b) a navigation path directed toward these waypoints. The DRL agent, which is pre-trained in our simulated environments, employs its learned policy to follow these guidance signals and control the robotic agent. In our experiments, the robot platform used is a ClearPath Husky Unmanned Ground Vehicle (UGV), and the semantic segmentation model employed is DACS [35].

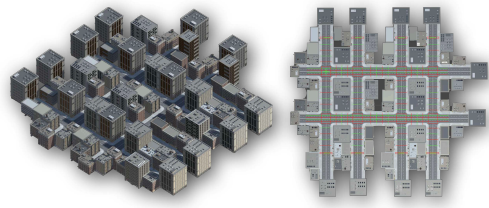


Fig. 5. An overview of the designed simulated environment, where the green and red lines represent the walking paths of the pedestrians.

#### V. EXPERIMENTAL RESULTS

In this section, we present the experimental results and discuss their implications. We first describe the experimental setups, followed by a comparison of various guidance schemes and a discussion of their impact. We then investigate the causes of failure cases and present the trajectories followed by the agents. Finally, we showcase a real-world experiment to demonstrate the innovative capability of generating virtual guidance from text prompts, as well as the successful transfer of the DRL agent's policy from virtual to real environments.

##### A. Experimental Setup

1) *Environment Setup*: To evaluate the efficacy of virtual guidance, we developed a virtual environment using Unity [19] and ML-Agents [32]. This environment was designed to emulate an urban landscape with eight intersections, as depicted in Fig. 5. The environment offers configurable starting and ending locations to create diverse routes for both training and evaluation of the agents. In addition, it incorporates dynamic objects with adjustable speeds, posing as obstacles that the agent needs to avoid while adhering to the provided virtual guidance. For the training phase, we employed a set of 89 routes. During the evaluation phase, we evaluated the agents under two separate scenarios: (a) *seen* routes and (b) *unseen* routes. The *seen* scenario focuses on the agent's capacity to follow virtual guidance and reach the destination without colliding with any obstacles. This set includes all 89 routes used during training. On the other hand, the *unseen* scenario aims to assess the agent's ability to adapt while navigating unfamiliar, novel routes with the support of virtual guidance. This scenario incorporates combinations of starting and ending points that were not included in the training phase, featuring four routes specifically selected for this purpose. This is essential for validating the applicability of the proposed method, as real-world scenarios could involve arbitrary planned routes, and it is not feasible to cover all possible routes during training.

2) *Agent Setup*: In our experiments, we implemented the DRL agent using a Deep Neural Network (DNN) and trained it with the Soft Actor-Critic (SAC) algorithm [36], [37]. The agent's observation space consists of three stacked semantic segmentation frames, each having dimensions of  $84 \times 180$ . These frames can be rendered either with or without virtual guidance. The frames are resized using bilinear downscaling, based on the outputs from the semantic segmentation

TABLE I  
AN IMPACT ANALYSIS OF THE DIFFERENT NAVIGATION GUIDANCE SCHEMES DISCUSSED IN THIS STUDY.

	Guidance Scheme	Representation Form	Performance				Failure Cases	
			SPL (↑)	Success Rate (↑)	Line Following Rate (↑)	Waypoint Collecting Rate (↑)	Collision Rate (↓)	Out-of-Bound (↓)
Seen	Hybrid (one-time)	{RGB, (r, θ)}	44.88 %	46.93 %	36.35 %	26.36 %	26.65 %	26.43 %
	VG <sub>waypoint</sub> (one-time)	RGB	73.16 %	73.51 %	69.54 %	73.53 %	19.41 %	6.95 %
	VG <sub>path</sub> (one-time)	RGB	72.68 %	72.75 %	89.46 %	✗	26.26 %	0.96 %
	VG <sub>path</sub> (real-time)	RGB	88.85 %	89.54 %	✗	✗	9.02 %	1.44 %
Unseen	Hybrid (one-time)	{RGB, (r, θ)}	18.96 %	20.86 %	35.33 %	20.33 %	37.02 %	42.12 %
	VG <sub>waypoint</sub> (one-time)	RGB	57.69 %	58.18 %	67.77 %	66.06 %	30.11 %	11.13 %
	VG <sub>path</sub> (one-time)	RGB	57.46 %	57.54 %	89.19 %	✗	36.94 %	5.45 %
	VG <sub>path</sub> (real-time)	RGB	81.48 %	82.74 %	✗	✗	15.08 %	1.99 %

model. The agent's action space is defined by a set  $\mathcal{A}$ , which includes two primary actions: **NOOP** and **TURN**( $\alpha$ ). The **NOOP** action maintains the agent's current directional orientation and advances it along a straight trajectory. On the other hand, the **TURN**( $\alpha$ ) action allows for incremental orientation adjustments, where the angle  $\alpha$  dictates the degree of the turn. The sign of  $\alpha$  is essential: negative values result in a leftward adjustment, while positive values prompt a rightward shift. This mechanism endows the agent with the ability to navigate and execute turns in a non-binary manner. The angular velocity  $\omega$ , influenced by these continuous adjustments to  $\alpha$ , is formulated as the following:

$$\omega += \alpha \times \kappa \times \Delta t, \quad (1)$$

where  $\Delta t$  represents the time interval, and  $\kappa$  is the steering sensitivity. We set  $\alpha$  to a standard value of  $35^\circ/\text{s}^2$  and  $\kappa$  to two. Rather than relying on abrupt, binary changes in direction, the agent's trajectory evolves based on the cumulative effects of successive adjustments to  $\alpha$ . Although the agent maintains a consistent velocity  $v$  of 6 m/s, its orientation continuously experiences refined and gradual modifications.

3) *Reward Function*: In the training phase of our experiments, we use the reward function that contains two terms: navigation following reward  $R_{nav}$  and goal reward  $R_{goal}$ . The navigation following reward  $R_{nav}$  is designed for encouraging the agent to follow the virtual guidance. For the *waypoints* scheme, the agent receives 5.0 when reaching any waypoint. While in the *navigation line* scheme, the reward is calculated based on the shortest distance between the agent's position and the navigation line, and is formulated as follows:

$$R_{nav} = \begin{cases} \min(\max(6 - \text{dist}(P_t, d'), 0.4)) & \text{if on the navigation line,} \\ -0.2 & \text{otherwise,} \end{cases} \quad (2)$$

where  $d'$  represents the shorted distance between the agent's position  $P_t$  with the navigation line at timestep  $t$ . For the goal reward  $R_{goal}$ , the agents receives 10.0 when reaching the destination, and  $-10.0$  if the agent collides with any obstacle, moves out the boundaries, or exceeds the time horizon. The final reward function  $R$  is calculated as  $R = R_{nav} + R_{goal}$ .

4) *Baseline*: To evaluate the effectiveness of the vision-based virtual guidance representation schemes described in Section III-B, we introduce a hybrid approach as a comparative baseline (denoted as 'Hybrid'). This approach combines

stacked segmentation observations with vector-based guidance to serve as the agent's inputs. In the vector provided to the agent, two critical parameters are encapsulated: the distance  $r$  to the closest forthcoming waypoint  $\omega^*$  and the orientation  $\Theta$ , which characterizes the agent's heading with respect to  $\omega^*$ . These are presented in polar coordinates as  $(r, \Theta)$ . The orientation  $\Theta$  falls within the range of  $-180$  to  $180$  degrees, i.e.,  $\Theta \in [-180^\circ, 180^\circ]$ . It should be noted that  $\Theta$  is normalized to a range of  $[-1, 1]$  before being offered to the agent. An example of this is visualized in Fig. 3 (e).

5) *Evaluation Metrics*: In our experiments, four metrics are utilized to evaluate the agent's performance from different perspectives. These four metrics are described as follows.

a) *Success rate*: In our evaluations, the success rate is utilized as a metric to assess the proficiency of the agent in reaching the designated destination.

b) *Success rate weighted by path length (SPL)*: The SPL metric evaluates the agent's navigational performance by accounting for both the success in reaching the destination and the efficiency of the selected trajectory [38]. This comparison ensures a consideration of both navigational success and efficiency. SPL is mathematically represented as follows:

$$\frac{1}{N} \sum_{i=1}^N \frac{l_i}{\max(l_i, p_i)}, \quad (3)$$

where  $l_i$  represents the shortest path distance from the agent's starting position to the goal for episode  $i$ , and  $p_i$  denotes the length of the path actually taken by the agent in that episode.

c) *Line following rate*: This metric evaluates the agent's proficiency in adhering to the virtual guidance path by calculating the overlapping ratio between the actual trajectory of the agent and the planned virtual guidance path.

d) *Waypoints collecting rate*: This metric calculates the ratio between the number of encountered waypoints and the total number of existing waypoints along the planned path.

## B. Impact Analysis of Virtual Guidance

This section presents a comparison of different guidance schemes. The virtual guidance schemes were previously discussed in Section III-B, while the baseline guidance scheme was discussed in Section V-A.4. The evaluation results, which include both the *seen* and *unseen* scenarios, are presented in Table I. It is worth noting that agents trained with waypoints are granted more flexibility in their

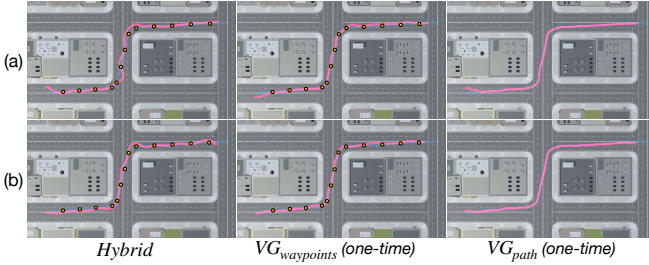













Fig. 6. Trajectory analysis comparing the navigation paths planned by the planning module with the actual paths navigated by the agent. The yellow balls denote the waypoints, the blue lines represent the one-time planned navigation paths, and the pink lines correspond to the actual trajectories.

TABLE II

COMPILATION OF TEXT PROMPTS EMPLOYED IN THE REAL-WORLD NAVIGATION EXPERIMENTS FOR VALIDATING VIRTUAL GUIDANCE.

Object Compositions	Text Prompts	Approaches	Success Rate
{  , 	{‘purple umbrella’ & ‘blue umbrella’}	$VG_{waypoints}$ $VG_{path} + VG_{waypoints}$	80.00 % 80.00 %
{  ,  , 	{‘purple umbrella’   ‘blue umbrella’   ‘green box’}	$VG_{waypoints}$ $VG_{path} + VG_{waypoints}$	80.00 % 60.00 %
{  ,  , 	{‘purple umbrella’   ‘blue box’   ‘green box’}	$VG_{waypoints}$ $VG_{path} + VG_{waypoints}$	60.00 % 80.00 %
{  ,  , 	{‘traffic cone’   ‘blue box’   ‘green box’}	$VG_{waypoints}$ $VG_{path} + VG_{waypoints}$	60.00 % 60.00 %

line following rate, as they can take any trajectory between two different waypoints. We evaluate the real-time and one-time planning settings described in Section III-B.1. The evaluation results presented in Table I indicate that the schemes incorporating virtual guidance (denoted as ‘ $VG$ ’) consistently outperform the baseline scheme (denoted as ‘*Hybrid*’) in terms of SPL, success rate, line following rate, and waypoint collection rate, in both *seen* and *unseen* scenarios. This suggests that our vision-based guidance strategies can effectively offer informative navigational cues. Therefore, it is able to alleviate the agent’s burden to learn the correlation between visual observations and navigational instructions derived from different non-visual modalities.

### C. Failure Case Analysis

To further examine the impact of different guidance schemes, we conducted an analysis of failure cases to identify their root causes. These cases are categorized into two types: (a) *out-of-bound (OOB)*, where the agent traverses into prohibited areas such as sidewalks, and (b) *collision*, which refers to instances where the agent collides with either static or dynamic obstacles. The results of this analysis are also presented in Table I. An observation from this analysis is that the agents trained with  $VG_{path}$  exhibit a lower OOB compared to those trained with  $VG_{waypoint}$  and *Hybrid*. A lower OOB rate implies that the agent is more likely to stay on the correct path while also avoiding prohibited areas. In contrast,  $VG_{waypoint}$  shows a lower collision rate than  $VG_{path}$ . This is because the agent has more flexibility when navigating between waypoints, which reduces the likelihood of collisions.  $VG_{path}$ , due to its reward structure that encourages path following, tends to stick more closely to the navigation path to earn rewards, which increases its chances of colliding

with pedestrians if they are on the virtual navigation path.

### D. Trajectory Analysis

To further investigate the rationale behind the observations mentioned above, we present our qualitative results in Fig. 6, where the one-time planned navigation paths are depicted as blue lines, the waypoints are depicted as yellow balls, and the actual trajectories followed by the agent are depicted in pink. We plotted the trajectories for successful cases from the agents trained using three configurations:  $VG_{path}$  (one-time),  $VG_{waypoints}$  (one-time), and *Hybrid*. It can be observed that the agents using  $VG_{path}$  closely adhere to the planned path, whereas the  $VG_{waypoints}$  agents gather the balls while maintaining some flexibility between waypoints. In contrast, the baseline *Hybrid* agents frequently deviates from the waypoints, resulting in erratic, circuitous trajectories. This may be due to the difficulties the agents face in reconciling different modalities, specifically visual observations and vector-based instructions, for this baseline.

### E. Real-World Validation of Virtual Guidance

As described in Section IV, we validated the proposed virtual guidance concept in real-world scenarios, using text prompts to derive virtual guidance paths. In our experimental configurations, four different compositions of objects were arranged, as summarized in Table II. Each composition contained multiple objects, with at least two objects from the same category but with distinct appearances. For example, the second composition featured two differently designed umbrellas along with a green box. The agent received a text prompt describing a particular composition of objects. Each experiment was performed over five independent runs. Owing to the capabilities of GroundingDINO, all objects described in the text prompts were correctly detected. The results are presented in Table II. The symbols & and | indicate that & requires the agent to reach all the objects described by the text prompt, while | requires the agent to reach just one of them. For example, in Fig. 1 (b), the agent first moves toward the purple umbrella and then proceeds to navigate to the blue umbrella. The agent’s task was to identify and approach the correct object(s) based on the provided text prompt, using the transformed virtual guidance. The results indicate that the policies trained in virtual environments can be effectively transferred to real-world settings, and that the agent can follow virtual guidance paths generated via text-prompt-based navigation objectives.

## VI. CONCLUSION

In this paper, we introduced the concept of virtual guidance and explored different potential methods of implementing it, including virtual navigation paths and virtual waypoints. A virtual city was created to train and evaluate the proposed virtual guidance scheme, where a planning module generated a path from randomly selected starting and ending points. The planned path was then rendered as virtual guidance to direct the agent. In our experiments, multiple evaluation metrics were employed to assess the effectiveness

of the proposed virtual guidance approaches. We compared the performance of the agents trained with virtual guidance to those trained with a baseline guidance scheme. Our results demonstrated that the agents trained with virtual guidance outperformed those trained with the baseline. This observation was also supported by our qualitative results. Finally, in our real-world evaluations, we extended virtual guidance by converting text prompts into visual cues. The experiments on a Husky AGV confirmed that the policies trained in virtual settings can effectively be transferred to real-world scenarios.

## REFERENCES

- [1] M. Mueller, A. Dosovitskiy, B. Ghanem, and V. Koltun, "Driving policy transfer via modularity and abstraction," in *Proc. Conf. on Robot Learning (CoRL)*, 2018, pp. 1–15.
- [2] S. Bansal, V. Tolani, S. Gupta, J. Malik, and C. Tomlin, "Combining optimal control and learning for visual navigation in novel environments," 2019.
- [3] J. Krantz, A. Gokaslan, D. Batra, S. Lee, and O. Maksymets, "Waypoint models for instruction-guided navigation in continuous environments," in *Proc. IEEE/CVF Int. Conf. on Computer Vision (ICCV)*, October 2021, pp. 15 162–15 171.
- [4] R. Guldenring, M. Görner, N. Hendrich, N. J. Jacobsen, and J. Zhang, "Learning local planners for human-aware navigation in indoor environments," in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2020, pp. 6053–6060.
- [5] A. Pokle *et al.*, "Deep local trajectory replanning and control for robot navigation," in *Proc. Int. Conf. on Robotics and Automation (ICRA)*, 2019, pp. 5815–5822.
- [6] C. Li, F. Xia, R. Martín-Martín, and S. Savarese, "HRL4IN: Hierarchical reinforcement learning for interactive navigation with mobile manipulators," in *Proc. Conf. on Robot Learning (CoRL)*, 2019, pp. 603–616.
- [7] A. Faust *et al.*, "PRM-RL: Long-range robotic navigation tasks by combining reinforcement learning and sampling-based planning," in *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2018, pp. 5113–5120.
- [8] L. Kästner *et al.*, "Connecting deep-reinforcement-learning-based obstacle avoidance with conventional global planners using waypoint generators," in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2021, pp. 1213–1220.
- [9] L. Kästner *et al.*, "Arena-Rosnav: Towards deployment of deep-reinforcement-learning-based obstacle avoidance into conventional autonomous navigation systems," in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2021, pp. 6456–6463.
- [10] J. Wöhlke, F. Schmitt, and H. van Hoof, "Hierarchies of planning and reinforcement learning for robot navigation," in *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2021, pp. 10 682–10 688.
- [11] B. Brito, M. Everett, J. P. How, and J. Alonso-Mora, "Where to go Next: Learning a subgoal recommendation policy for navigation in dynamic environments," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 4616–4623, 2021.
- [12] L. Kästner, X. Zhao, Z. Shen, and J. Lambrecht, "A hybrid hierarchical navigation architecture for highly dynamic environments using time-space optimization," in *Proc. IEEE/SICE Int. Symp. on System Integration (SII)*, 2023, pp. 1–8.
- [13] D. S. Chaplot, R. Salakhutdinov, A. Gupta, and S. Gupta, "Neural topological SLAM for visual navigation," in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 12 872–12 881.
- [14] M. Mueller, A. Dosovitskiy, B. Ghanem, and V. Koltun, "Driving policy transfer via modularity and abstraction," in *Proc. Conf. on Robot Learning (CoRL)*, 2018, pp. 1–15.
- [15] A. Sax *et al.*, "Learning to navigate using mid-level visual priors," in *Proc. Conf. on Robot Learning (CoRL)*, 2020, pp. 791–812.
- [16] B. Chen *et al.*, "Robust policies via mid-level visual representations: An experimental study in manipulation and navigation," in *Proc. Conf. on Robot Learning*, 2021, pp. 2328–2346.
- [17] H.-K. Yang *et al.*, "Investigation of factorized optical flows as mid-level representations," in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2022, pp. 746–753.
- [18] Z.-W. Hong *et al.*, "Virtual-to-Real: Learning to control in visual semantic segmentation," in *Proc. Int. Joint Conf. on Artificial Intelligence IJCAI*, 2018, pp. 4912–4920.
- [19] Unity Technologies, "Unity engine," <https://unity.com>.
- [20] K. Yadav, J. Krantz, R. Ramakranya, S. K. Ramakrishnan, J. Yang, A. Wang, J. Turner, A. Gokaslan, V.-P. Berges, R. Mootaghi, O. Maksymets, A. X. Chang, M. Savva, A. Clegg, D. S. Chaplot, and D. Batra, "Habitat challenge," 2023.
- [21] S. Liu, Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang, C. Li, J. Yang, H. Su, J. Zhu, *et al.*, "Grounding dino: Marrying dino with grounded pre-training for open-set object detection," *arXiv preprint arXiv:2303.05499*, 2023.
- [22] S. Gupta, J. Davidson, S. Levine, R. Sukthankar, and J. Malik, "Cognitive mapping and planning for visual navigation," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 7272–7281.
- [23] Y. Liang, B. Chen, and S. Song, "SSCNav: Confidence-aware semantic scene completion for visual semantic navigation," in *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2021, pp. 13 194–13 200.
- [24] S. Bansal, V. Tolani, S. Gupta, J. Malik, and C. Tomlin, "Combining optimal control and learning for visual navigation in novel environments," in *Proc. Conf. on Robot Learning (CoRL)*, 2020, pp. 420–429.
- [25] W. Gao, D. Hsu, W. S. Lee, S. Shen, and K. Subramanian, "Intention-Net: Integrating planning and deep learning for goal-directed autonomous navigation," in *Proc. Conf. on Robot Learning*, 2017, pp. 185–194.
- [26] A. Mousavian *et al.*, "Visual representations for semantic target driven navigation," in *Proc. Int. Conf. on Robotics and Automation (ICRA)*, 2019, pp. 8846–8852.
- [27] Y. Zhu *et al.*, "Target-driven visual navigation in indoor scenes using deep reinforcement learning," in *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2017, pp. 3357–3364.
- [28] K. Fang, A. Toshev, L. Fei-Fei, and S. Savarese, "Scene memory transformer for embodied agents in long-horizon tasks," in *Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 538–547.
- [29] D. S. Chaplot, D. Gandhi, A. Gupta, and R. Salakhutdinov, "Object goal navigation using goal-oriented semantic exploration," in *Proc. Int. Conf. on Neural Information Processing Systems (NeurIPS)*, 2020.
- [30] S. K. Ramakrishnan, D. S. Chaplot, Z. Al-Halah, J. Malik, and K. Grauman, "PONI: Potential functions for objectgoal navigation with interaction-free learning," in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 18 868–18 878.
- [31] W. Zhao, J. P. Queralta, and T. Westerlund, "Sim-to-real transfer in deep reinforcement learning for robotics: a survey," in *SSCI*, 2020.
- [32] A. Juliani, V.-P. Berges, E. Teng, *et al.*, "Unity: A general platform for intelligent agents," *arXiv preprint arXiv:1809.02627*, 2020.
- [33] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Trans. on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968.
- [34] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.
- [35] W. Tranheden, V. Olsson, J. Pinto, and L. Svensson, "Dacs: Domain adaptation via cross-domain mixed sampling," in *Proc. IEEE/CVF Winter Conf. on Applications of Computer Vision (WACV)*, January 2021, pp. 1379–1389.
- [36] T. Haarnoja *et al.*, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *ICML*, 2018.
- [37] P. Christodoulou, "Soft actor-critic for discrete action settings," *arXiv preprint arXiv:1910.07207*, 2019.
- [38] P. Anderson, A. X. Chang, D. S. Chaplot, A. Dosovitskiy, S. Gupta, V. Koltun, J. Kosecka, J. Malik, R. Mottaghi, M. Savva, and A. R. Zamir, "On evaluation of embodied navigation agents," *arXiv preprint arXiv:1807.06757*.