Integrating Visual and Linguistic Instructions for Context-Aware Navigation Agents

Suhwan Choi[*] MAUM.AI

Yongjun Cho[*] MAUM.AI

Minchan Kim[*] MAUM.AI

Jaeyoon Jung[*] MAUM.AI

Myunchul Joe MAUM.AI

Yubeen Park MAUM.AI

Minseo Kim Yonsei University

Sungwoong Kim Yonsei University

Sungjae Lee Yonsei University

Hwiseong Park MAUM.AI

Jiwan Chung Yonsei University

Youngjae Yu Yonsei University

Abstract

Real-life robot navigation involves more than just reaching a destination; it requires optimizing movements while addressing scenario-specific goals. An intuitive way for humans to express these goals is through abstract cues like verbal commands or rough sketches. Such human guidance may lack details or be noisy. Nonetheless, we expect robots to navigate as intended. For robots to interpret and execute these abstract instructions in line with human expectations, they must share a common understanding of basic navigation concepts with humans. To this end, we introduce CANVAS, a novel framework that combines visual and linguistic instructions for commonsense-aware navigation. Its success is driven by imitation learning, enabling the robot to learn from human navigation behavior. We present COMMAND, a comprehensive dataset with human-annotated navigation results, spanning over 48 hours and 219 km, designed to train commonsense-aware navigation systems in simulated environments. Our experiments show that CANVAS outperforms the strong rule-based system ROS NavStack across all environments, demonstrating superior performance with noisy instructions. Notably, in the orchard environment, where ROS NavStack records a 0% total success rate, CANVAS achieves a total success rate of 67%. CANVAS also closely aligns with human demonstrations and commonsense constraints, even in unseen environments. Furthermore, real-world deployment of CANVAS showcases impressive Sim2Real transfer with a total success rate of 69%, highlighting the potential of learning from human demonstrations in simulated environments for real-world applications.

1 Introduction

Real-life robot navigation scenarios involve addressing complex objectives that extend far beyond simply reaching a destination. For example, an agricultural spraying robot must maximize field coverage [24], while a package delivery robot must adhere to road lanes and use crosswalks when transitioning between sidewalks. [37, 12] In both cases, robots need to optimize their movements while responding to the specific requirements of the scenario.

Humans typically communicate these scenario-specific goals through high-level guidance, such as verbal commands [1, 18, 44], rough sketches of the desired route [33, 4], or a combination of

[*] Equal Contribution. Videos, datasets, and models: worv-ai.github.io/canvas

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

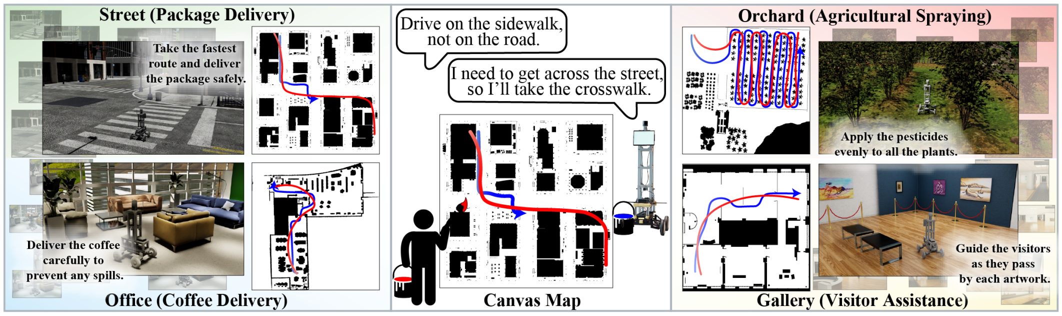

Street (Package Delivery)
Take the fastest route and deliver the package safely.
Drive on the sidewalk, not on the road.
I need to get across the street, so I'll take the crosswalk.
Orchard (Agricultural Spraying)
Apply the pesticides evenly to all the plants.
Deliver the coffee carefully to prevent any spills.
Office (Coffee Delivery)
Canvas Map
Guide the visitors as they pass by each artwork.
Gallery (Visitor Assistance)

Figure 1: Humans often give abstract navigation directions using simple instruction, relying on the recipient's commonsense to bridge the gaps. With CANVAS, robots can interpret and act on these directions like humans do, fostering a shared understanding of the environment. It shows how robots can use commonsense to translate vague human instructions into concrete actions, navigating across diverse settings in our COMMAND dataset, which we plan to open-source as valuable resources for imitation learning in commonsense-aware navigation.

both [47]. While such guidance outlines the robot's overall objectives, it often lacks the specificity required for precise execution. To convert these abstract and imprecise instructions into actionable navigation plans, robots need commonsense knowledge. In the context of robotics, commonsense refers to the general understanding humans naturally use to make decisions, covering aspects such as human desires, physics, and causality [36]. Robots must leverage this knowledge to flexibly adjust their paths, ensuring their decisions align with human expectations by adhering to commonsense constraints posed by the environment and the user's true intentions.

In response to these challenges, we introduce CANVAS (Commonsense-Aware NaVigAtion System), a novel framework for integrating abstract human instructions into robot navigation. Our approach utilizes both visual and linguistic inputs, such as rough sketch trajectories on map images or textual descriptions. These multimodal instructions are processed by a vision-language model that generates incremental navigation targets. By leveraging the commonsense knowledge embedded in pre-trained vision-language models [8, 46, 45, 7], robots can develop a versatile understanding of commonsense navigation dynamics. Quantifying successful navigation behaviors using rewards[38, 41, 35] is particularly difficult in commonsense-aware navigation. Therefore, we employ imitation learning, enabling the robot to comprehend user intent behind noisy and imprecise instructions from human demonstrations. [11]

Additionally, we introduce COMMAND (COMMonsense-Aware Navigation Dataset), a comprehensive dataset designed to train commonsense-aware navigation robots. The dataset features three simulated environments with distinct characteristics (office, street, and orchard). To facilitate imitation learning for instruction-following tasks, we provide 3,343 fully human-annotated navigation results from the simulated environments. Notably, COMMAND offers 48 hours of driving data, which is nearly three times longer than GoStanford [17], covering 219 km and thereby enriching the dataset's diversity and scope. Furthermore, we propose two metrics to evaluate the commonsense adherence of navigation algorithms: Trajectory Deviation Distance (TDD) and Instruction Violation Rate (IVR).

Our results show that CANVAS consistently outperforms ROS NavStack [28] across all environments with noisy sketch instructions. Particularly in the challenging orchard environment, CANVAS navigates effectively, while ROS NavStack fails due to its reliance on rule-based algorithms[9]. CANVAS 's trajectory closely mirrors human demonstrations with fewer commonsense constraint violations, indicating a better understanding of human expectations. Despite being trained only on simulated data, CANVAS also excels in real-world scenarios, demonstrating strong Sim2Real transfer capabilities.

Our contributions are threefold:

1. We introduce CANVAS, a novel framework that allows humans to easily communicate with robots using multimodal inputs, ensuring that robots effectively achieve navigation goals, even when human instructions are vague or noisy.

2. We introduce COMMAND, a dataset for training commonsense-aware navigation robots, featuring 48 hours of driving data over 219 kilometers, with fully human-annotated sketch instruction and navigation outcomes.
3. We present extensive experiments demonstrating that CANVAS outperforms ROS NavStack in success rate, collision rate, trajectory deviation distance, and instruction violation rate. To support further research, we are open-sourcing CANVAS and COMMAND for imitation learning in commonsense-aware robot navigation.

2 Related Work

2.1 Robot Navigation

Historically, robot navigation systems were largely rule-based [16, 9, 25], relying on a set of predefined rules, as seen in frameworks like the ROS NavStack [28]. Following the successful application of deep learning to robotics, more flexible neural navigation approaches have emerged. Visual navigation models, such as NoMaD [34], ViNT [32], and GNM [31], utilize images as goal representations. However, since their high-level planning heavily relies on the topological graph with first-person visual observations, they cannot handle unvisited locations and are sensitive to environmental changes. [3] Vision-language navigation integrate visual information from sensors with language instructions [14, 18, 44]. However, the ambiguous nature of language instructions poses limitations on controlling detailed navigation routes. [39] Recently, LIM2N [47] enabled users to control robots through natural language and sketch trajectories, combining high-level goals with precise motion paths for more intuitive interaction. However, the system's demand for highly accurate and detailed instructions, coupled with its vulnerability to missing details or small mistakes, reduces its usability for non-expert users and limits its effectiveness in a broader range of real-world applications [1]. Our proposed method, CANVAS, differentiates itself by addressing the challenge of interpreting abstract or noisy human instructions. CANVAS converts visual and linguistic instructions into detailed navigation actions, utilizing commonsense knowledge to fill in the gaps. This integration of commonsense enables CANVAS to dynamically adapt its navigation strategies across diverse contexts, resulting in enhanced task execution compared to ROS NavStack.

	***NoMaD* [34]**	***NaVid* [44]**	***LIM2N* [47]**	**CANVAS**
Instruction	Image	Language	Sketch, Language	Sketch, Language
Misleading (3.1)	✗	✗	✗	✓
Custom Dataset	✗	✓	✓	✓
Environment	Real	Real	Real, Simulation	Real, Simulation
Scenes	Indoor, Outdoor	Indoor	Indoor	Indoor, Outdoor

Table 1: Comparison between various robot navigation methods.

2.2 Imitation Learning in Robotics

Imitation Learning (IL) enables agents to learn tasks by mimicking expert demonstrations, eliminating the need for predefined rules or reward functions typically required in Reinforcement Learning (RL) [19]. By directly leveraging expert behavior, IL has proven particularly advantageous in situations where designing a reward function is challenging or exploration involves potential risks [42]. As a result, there has been a growing interest in applying IL to robot navigation [20, 29]. A key challenge in IL is modeling multimodal action distributions [30]. One solution is to quantize actions into discrete tokens, simplifying the action space [26, 10, 30, 6, 22]. Autoregressive prediction of quantized actions effectively reduces the complexity of modeling diverse and feasible action sequences. CANVAS builds upon this idea by converting continuous waypoints into 128 discrete waypoint tokens using K-means clustering [5, 23]. This approach enhances the ability of robots to model multimodal action distributions, enabling robust navigation strategies that adapt to diverse human instructions and environmental variations.

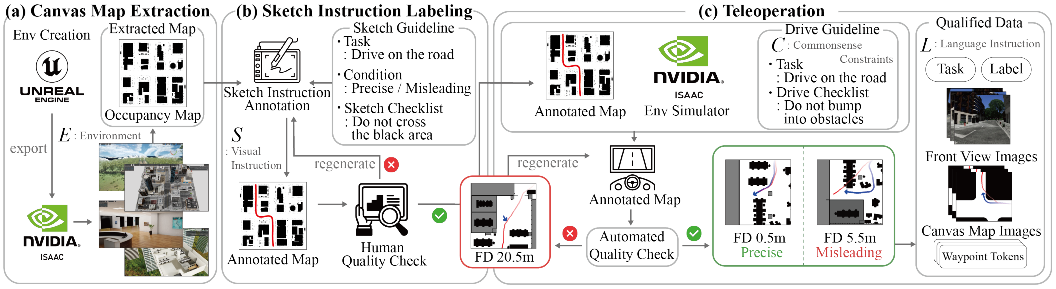


Figure 2: Data collection pipeline for COMMAND dataset. (a) First, we create diverse navigation environments and extract maps. (b) Then, human annotators sketch routes on the maps based on the guidelines. (c) Finally, we use teleoperation to collect human demonstrations. Red line shows the roughly sketched routes, while the blue line shows the optimal trajectory driven by the human annotator. FD refers to *Frechet distance*.

3 Dataset and Task

An interactive robot navigation framework should fulfill two key objectives: first, humans should be able to communicate desired routes and requirements intuitively; second, the robot should accurately interpret and execute those instructions. However, achieving these goals can be challenging. Simplifying communication for humans often complicates it for robots because humans naturally assume the listener shares their commonsense knowledge. This commonsense allows humans to infer meaning even when instructions are incomplete or imprecise but also causes robots to struggle without explicit and precise input.

To address this challenge, we introduce COMMAND—a comprehensive experiment suite designed to assess whether robots can use commonsense understanding to transform abstract or occasionally noisy human instructions into the most desired trajectory. COMMAND was collected in three distinct environments: office, street, and orchard, using NVIDIA Isaac Sim [27]. The data consists of high-quality sketch instruction labels and teleoperation data, **all by human experts**. COMMAND contains 48 hours of driving data covering a distance of 219 kilometers. In this section, we describe the dataset curation process and the task definition that builds upon it.

3.1 Dataset

A datapoint in COMMAND includes a canvas map, sketch and language instructions, commonsense constraints, and teleoperation records. The canvas maps, linked to each environment, not only provide occupancy information but also serve as a communication interface between humans and robots. We assume humans provide instructions in two modalities: sketch instructions S that consist of hand-drawn trajectories on maps for the robot to follow, and language instructions L that outline goals and related requirements. When collecting sketch instructions S, we introduce both ***Precise*** and ***Misleading*** conditions to gather training data that enables our model to handle noisy sketch instructions more robustly. In addition to the instructions, we define a set of commonsense constraints C, which are derived from the navigation environment E and the language instructions L. These constraints help evaluate whether the robot exhibits appropriate navigation behaviors, such as using crosswalks. We also include human teleoperation records to optimize and evaluate the robot behavior against human actions. Data curation process for COMMAND involved one month of full-time effort by an expert designer, alongside 25 trained data workers. Before participation, all data workers completed a preparatory course on the collection pipeline. Quality control was ensured through weekly random sample checks by a manager. An overview of our data collection pipeline is illustrated in Figure 2, and the statistics are provided in Table 2. We detail each step below.

Environments. COMMAND features three simulated environments, each tailored to specific scenarios. The simulated environments include tasks such as coffee delivery in an office, package delivery on the street, and agricultural spraying in an orchard. An expert designer creates each simulated environment using Unreal Engine [13].

Canvas Map Extraction. The simulated environments are subsequently exported to NVIDIA Isaac Sim, where occupancy maps are extracted programmatically.

Sketch Instruction Labeling. The data workers draw sketch trajectories on the canvas maps following sketch guidelines manually crafted by the authors. When the *Precise* condition is included, the workers trace the most efficient route, closely following the guidelines. In contrast, when the *Misleading* condition is included, data workers are instructed to deliberately introduce noise by drawing trajectories that pass through walls or objects. All sketch instructions undergo manual inspection to ensure quality.

Teleoperation. The data workers are then provided both the sketch and language instructions to teleoperate the virtual robot in NVIDIA Isaac Sim. We record front view images and canvas map images along with the sketch trajectories. After collecting the teleoperation data, we adopt the *Frechet distance* (FD) [2] to measure the discrepancy between the sketch trajectory and the human-demonstrated trajectory. This metric, which indicates noise in the sketch instructions, tends to be higher in the *Precise* condition and lower in the *Misleading* condition. We also conducted an automated quality check, filtering out outliers with unusually high FD values.

Split	Train				Test					
Environment	Office	Street		Orchard	Office	Street		Orchard	Gallery	Real Office
		Road	Sidewalk			Road	Sidewalk			
Count	2,263	403	410	267	10	20	10	10	10	10
Avg. Time	31s	57s	103s	172s	39s	56s	127s	182s	76s	30s
Avg. Distance	32.8m	80.4m	150.0m	191.6m	38.7m	91.0m	152.0m	232.88m	48.8m	16.0m
Avg. FD (P / M)	1.05 / 1.77	0.97 / 2.02	3.03 / 3.50	1.91 / 3.76	0.77 / 1.62	1.28 / 1.65	1.32 / 2.33	1.51 / 2.27	0.68 / 1.36	1.44 / 1.63
% of Misleading	31%	51%	51%	40%	50%	50%	50%	50%	50%	50%

Table 2: Statistics for the training and test set. Training dataset includes 48 hours of driving data over 219 kilometers, while the test dataset consists of 1.6 hours. FD refers to *Frechet distance*, where (P / M) stands for Precise and Misleading. The unit of FD is meters.

3.2 Task Definition

Our visual instructions S consist of hand-drawn trajectories on maps for the robot to follow, while the linguistic instructions L outline goals and related requirements. In addition to the instructions, we define a set of commonsense constraints C, which are derived from the navigation environment E and the language instructions L. The robot R is tasked with following the trajectory indicated by S while adhering to the commonsense constraints C. The robot R is tasked with executing a sequence of actions to complete the instructions provided by the human. It must not only follow the trajectory indicated by S, but also adhere to the commonsense constraints C.

At each timestep t, the robot R manages two states: the front view image $X_f(t)$ and the robot's hindsight trajectory [15] up to timestep $t-1$, denoted as $H(t)$. The front view image $X_f(t)$ is captured by the robot's camera, while $H(t)$ is tracked through odometry to log the robot's past positions. We combine the sketch instruction S and hindsight trajectory $H(t)$ onto the same map to create the canvas map image $X_c(t)$. At each step, the robot R generates an action $y(t) = [w_0, w_1, w_2, w_3]$, which is a sequence of waypoints. This action is conditioned on the front view image $X_f(t)$, the canvas map image $X_c(t)$, and the language instruction L. Formally, the robot's action is defined as:

$$y(t) = R(X_f(t), X_c(t), L)$$

At the end of each iteration, the hindsight trajectory is updated by appending $p(t)$, which represents the robot's position updated through predicted waypoints $y(t)$, resulting in $H(t+1) = (p(1), p(2), ..., p(t))$. This update is reflected in the next canvas map image $X_c(t+1)$, while the front view image $X_f(t+1)$ is also updated based on the robot's new position. This process continues until the robot either reaches the destination (within 0.5 meters) or a maximum timestep $t = T$ is reached.

4 Method

To address the problem formulated in Section 3, we introduce CANVAS, a navigation system designed to bridge the gap between abstract human instructions and concrete robot actions by leveraging commonsense understanding from pre-trained vision-language models (VLMs) [21]. In this section, we provide an overview of the model architecture, as well as the training and inference.

4.1 Architecture

The model architecture is illustrated in Figure 3. We adopt a VLM denoted as π_θ. The front view image $X_f(t)$ and canvas map image $X_c(t)$, which is cropped to an 8-meter radius around the robot, are processed through a vision encoder $g_\phi(\cdot)$. This results in two visual features, $Z_f = g_\phi(X_f(t))$ and $Z_c = g_\phi(X_c(t))$. A projector p_ϕ is used to project these visual features into the word embedding space, producing a sequence of visual tokens $\tau_v = p_\phi(Z_f, Z_c)$. A sequence of language tokens τ_l is also obtained from the language instruction L. Both the visual tokens τ_v and language tokens τ_l are then fed into the large language model denoted as $f_\phi(\cdot)$, which outputs the waypoint tokens $[w_0, w_1, w_2, w_3] = f_\phi(\tau_v, \tau_l)$. Due to the reasons mentioned in Section 2.2, we apply the simplest method, K-means clustering, to discretize continuous waypoints into tokens, with empirical testing showing that K=128 outperformed 32, 64, or 256. Fewer tokens can hinder precise actions like navigating narrow passages.

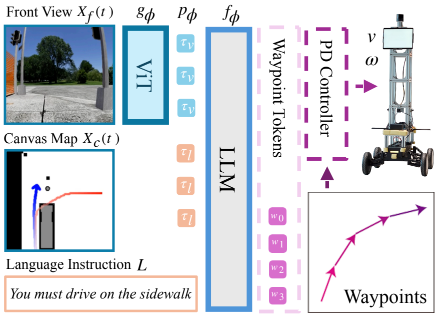


Figure 3: Overview of the CANVAS framework. It processes the front view image $X_f(t)$, canvas map $X_c(t)$, and language instruction L to generate waypoint tokens, which are passed to a PD controller to move the robot.

4.2 Training

CANVAS is designed to generate actions as a sequence of waypoint tokens. A robot's trajectory can be represented by N consecutive waypoints. During training, we minimize the negative log-likelihood loss, which is formulated as follows:

$$J(\pi_\theta) = \sum_{n=1}^{N} \sum_{t=0}^{3} \log \pi_\theta \left(w_t^n \mid X_f(t)^n, X_c(t)^n, L^n \right)$$

The model reframes the navigation as a classification problem, where it predicts the next waypoint based on the current state and given instructions. As explained in Section 2.2, by treating navigation as a classification task, the model can manage multimodal distributions, enhancing both stability and accuracy in complex environments.

4.3 Inference

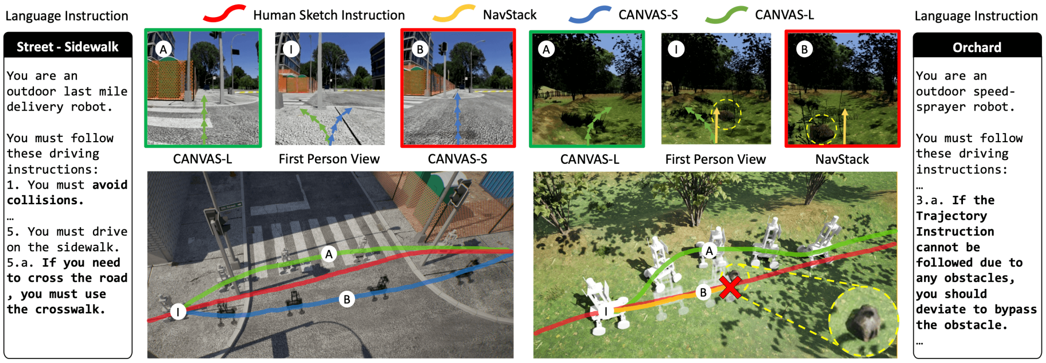
During inference, the model-generated discrete waypoint tokens convert into continuous waypoints, which are then input into a Proportional-Derivative (PD) controller to produces linear and angular velocities (v, ω) for the robot's actuators.

5 Experiments

In this section, we aim to answer the following questions:

A. Can CANVAS handle the commonsense-aware navigation tasks in simulated environments?
B. Can CANVAS be transferred to a real-world environment in a zero-shot manner?
C. How much does leveraging the pre-trained weights of the VLM enhance performance?
D. When NavStack fails, in what ways can CANVAS succeed?
E. Is CANVAS fast enough for real-time navigation?

Experimental Setup. In COMMAND, successful navigation requires the robot to reach the target location without collisions, while also respecting commonsense constraints. As a result, four key metrics were used to evaluate performance: SR, CR, TDD, and IVR. Success Rate (SR) represents the proportion of successful episodes, while Collision Rate (CR) captures the proportion of episodes with collisions. Trajectory Deviation Distance (TDD) measures how closely the model follows human demonstrations, using the interquartile mean of *Frechet distances*. Finally, Instruction Violation Rate (IVR) assesses the proportion of episodes where human evaluators identified violations of



Street - Sidewalk

You are an outdoor last mile delivery robot.

You must follow these driving instructions:
1. You must **avoid collisions.**
…
5. You must drive on the sidewalk.
5.a. **If you need to cross the road , you must use the crosswalk.**

Orchard

You are an outdoor speed-sprayer robot.

You must follow these driving instructions:
…
3.a. **If the Trajectory Instruction cannot be followed due to any obstacles, you should deviate to bypass the obstacle.**
…

Figure 4: The left side of the figure compares CANVAS-L and CANVAS-S, showing CANVAS-L using the crosswalk despite a misleading sketch instruction. The right side compares CANVAS-L and NavStack, illustrating CANVAS-L avoiding small obstacles, such as rocks.

Method	Precise			Misleading			Total
	SR(↑)	CR(↓)	TDD(↓)	SR(↑)	CR(↓)	TDD(↓)	SR(↑)
Seen Environment							
Office							
NavStack	87%	13%	0.846m	0%*	100%*	-	-
CANVAS-S	**100%**	**0%**	**0.730m**	87%	13%	0.843m	93%
CANVAS-L	**100%**	**0%**	0.802m	**100%**	**0%**	**0.753m**	**100%**
Street (Road)							
NavStack	**100%**	**0%**	1.654m	0%*	100%*	-	-
CANVAS-S	**100%**	**0%**	1.189m	**100%**	**0%**	**1.075m**	**100%**
CANVAS-L	97%	3%	**1.117m**	97%	3%	1.236m	97%
Street (Sidewalk)							
NavStack	53%	53%	1.450m	0%*	100%*	-	-
CANVAS-S	60%	40%	1.451m	47%	53%	2.379m	54%
CANVAS-L	**87%**	**13%**	**1.394m**	**53%**	**47%**	**1.839**m	**70%**
Orchard							
NavStack	0%	87%	-	0%*	100%*	-	-
CANVAS-S	**73%**	60%	**1.561m**	**60%**	**33%**	1.448m	**67%**
CANVAS-L	67%	**47%**	1.759m	**60%**	53%	**1.392m**	64%
Unseen Environment							
Gallery							
NavStack	**100%**	**0%**	0.783m	0%*	100%*	-	-
CANVAS-S	87%	13%	**0.773m**	**33%**	**66%**	0.938m	60%
CANVAS-L	**100%**	7%	0.9m	**33%**	**66%**	**0.856m**	**67%**

(a) Evaluation results on simulated environments.

Environment	Method	Precise IVR(↓)	Misleading IVR(↓)
Street (Road)	NavStack	7%	100%*
	CANVAS-S	**0%**	**7%**
	CANVAS-L	17%	30%
Street (Sidewalk)	NavStack	7%	100%*
	CANVAS-S	**0%**	26%
	CANVAS-L	**0%**	**13%**

(b) Evaluation of violation rates for commonsense constraints in a street environment.

Method	Precise SR(↑)	Misleading SR(↑)	Total SR(↑)
NavStack	**100%**	0%*	-
CANVAS-S	77%	**60%**	**69%**
CANVAS-L	93%	33%	63%

(c) Evaluation results on real environments.

Table 3: Evaluation results on both simulated and real environments. *: NavStack was not tested in the misleading scenario because it is not equipped to handle such situations.

commonsense constraints, such as keeping to the right lane or using crosswalks. TDD was calculated only for success cases, as including failure cases would skew the metric.

We compare CANVAS with **ROS NavStack** [28], a straightforward yet effective rule-based navigation system. For this system, we converted the sketch instructions into step-by-step, point-to-point inputs, but language instructions could not be accommodated. The same hyperparameters were used for all experiments with NavStack. We evaluate two variations of CANVAS. **CANVAS-S** modifies the original Idefics2 8B [21] by swapping the vision encoder for SigLIP-L [43] and the text encoder for Qwen2-0.5B [40], reducing the model size from 8B to 0.7B to better accommodate real-world deployment. In contrast, **CANVAS-L** retains the original Idefics2 8B [21] architecture with its pre-trained weights. Both models utilize 128 waypoint tokens.

In the main experiments, both models were inferred using single NVIDIA H100 GPU. All experiments were evaluated over three iterations for each test dataset, with randomized starting orientations. In the real-world environment, SLAM with FAST-LIO2 was used to find the robot's current position.

5.1 Results in the Simulated Environments

Seen Environments. Table 3a shows CANVAS's performance in three seen environments: office, street (road, sidewalk), and orchard. Under precise instructions, CANVAS achieves similar SR and CR to NavStack in the office and street (road), where navigation is easier, indicating that CANVAS can learn the essential navigation behaviors effectively from human demonstrations. However, in more challenging environments like the street (sidewalk) and orchard, CANVAS significantly outperforms NavStack. A detailed analysis of CANVAS's performance is provided in Section 5.4. For instance, in the orchard, where the LiDAR of NavStack struggles with obstacle detection, CANVAS uses camera inputs to recognize obstacles and assesses the risk, leading to better navigation. Under misleading instructions, NavStack suffers with 0% SR and 100% CR due to rigid trajectory adherence, while CANVAS maintains consistent performance in terms of TDD, showing adaptability in finding the optimal trajectory despite noisy guidance.

Table 3b compares the IVR between CANVAS and NavStack. In the road, commonsense constraints include lane adherence, while in the sidewalk, they involve crossing roads correctly and staying on the sidewalk. CANVAS consistently achieves lower IVR, particularly under misleading instructions, by learning driving rules from demonstrations, unlike NavStack's reliance on explicit programming.

Unseen Environment. We exclude the gallery environment during training to evaluate CANVAS's performance in unseen settings. As demonstrated in Table 3a, CANVAS continues to show strong navigation capabilities, even in scenarios with noisy guidance. This demonstrates that CANVAS 's navigation capability generalizes well to unseen environments.

5.2 Results in the Real-World Environment

While COMMAND collects human demonstrations across a variety of simulated environments designed to resemble real-world conditions, a potential concern is whether these simulations fully capture the complexity of the real world. Therefore, it is important to demonstrate that the CANVAS's effective navigation in simulation can extend to real-world environments. To assess its real-world performance, we tested CANVAS in an actual office environment that was used as the basis for the simulation. Despite being trained solely on simulated data, CANVAS demonstrated strong Sim2Real transfer capabilities, performing reliably in real-world scenario.

5.3 Ablation Study

We explore the importance of leveraging pre-trained weights from the vision-language models. As demonstrated in Table 4, these weights were crucial for CANVAS's performance, especially in both unseen and real-world settings. This indicates that the knowledge encapsulated in the pre-trained VLMs offered a strong foundation to learn how to incorporate them in developing a generalizable understanding of driving dynamics.

Environment	Method	Precise SR(↑)	Misleading SR(↑)	Total SR(↑)
Seen - Office	CANVAS-L	**100%**	**100%**	**100%**
	w/o PT	**100%**	87%	93%
Unseen - Gallery	CANVAS-L	**100%**	33%	**67%**
	w/o PT	60%	**40%**	50%
Real - Office	CANVAS-L	**93%**	**33%**	**63%**
	w/o PT	73%	**33%**	53%

Table 4: Ablation study on the effect of VLM pre-training. PT refers to the pre-trained weights.

5.4 Additional Case Study

We perform a qualitative analysis to examine the factors behind the success of CANVAS in comparison to NavStack [28]. Figure 5 highlights typical failure cases for NavStack, where the robot is unable to reach its destination. In 56% of these cases, failures result from stumbling over rocks in the orchard environment. Figure 4 compares CANVAS and NavStack in this primary failure scenario. The orchard has uneven terrain, and NavStack struggles to avoid small but hazardous obstacles like rocks because its limited perception can't distinguish between rocks and passable areas such as grass. In contrast, CANVAS utilizes visual inputs from the camera to reliably detect unexpected obstacles and, through experience learned from demonstrations, evaluates their risk to navigation. Additionally, we assess CANVAS-S and CANVAS-L on their adherence to commonsense constraints. As shown in

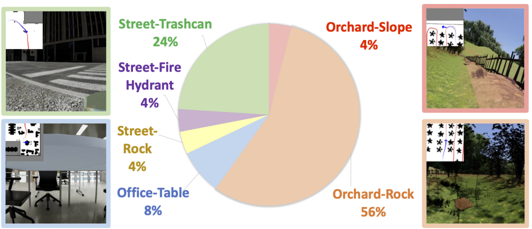


Figure 5: We classify the failure cases of NavStack [28] in various simulated environments.

Figure 4, CANVAS-S disregards the use of a crosswalk, whereas CANVAS-L successfully adheres to the implicit commonsense rule of crossing at designated points.

5.5 Real-Time Navigation Feasibility Study

Finally, we evaluate the feasibility of deploying CANVAS in real-time applications. CANVAS demonstrates real-time inference capabilities, with an average latency of 400ms for CANVAS-S and 800ms for CANVAS-L, all within the available memory limits. These results highlight CANVAS's potential to efficiently handle real-world navigation tasks without substantial delays.

6 Conclusion

We present CANVAS, a novel commonsense-aware navigation system that learns from human demonstrations through imitation learning. CANVAS allows intuitive human instructions using abstract sketches and natural language while leveraging commonsense reasoning to bridge the gap between vague human guidance and concrete robot actions. With the COMMAND dataset for imitation learning and pre-trained vision-language models, CANVAS allows robots to understand implicit human intent and make decisions aligned with human expectations. Experiments show that CANVAS outperforms ROS NavStack, a strong rule-based system, with higher success rates, fewer collisions, and better trajectory alignment with human demonstrations, all while adhering to commonsense constraints. Additionally, CANVAS exhibits strong performance in both unseen and real-world environments, highlighting its generalization capabilities. By open-sourcing the COMMAND dataset and CANVAS, we hope to contribute to active research on imitation learning techniques for commonsense reasoning in robot navigation.

Acknowledgments

This research was partly supported by GINT. The authors would like to thank GINT for providing feedback on the agricultural spraying robot.

References

[1] P. Anderson, Q. Wu, D. Teney, J. Bruce, M. Johnson, N. Sünderhauf, I. Reid, S. Gould, and A. 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 (CVPR)*, 2018.

[2] B. Aronov, S. Har-Peled, C. Knauer, Y. Wang, and C. Wenk. Fréchet distance for curves, revisited. In *Algorithms–ESA 2006: 14th Annual European Symposium, Zurich, Switzerland, September 11-13, 2006. Proceedings 14*, 2006.

[3] F. Blochliger, M. Fehr, M. Dymczyk, T. Schneider, and R. Siegwart. Topomap: Topological mapping and navigation based on visual slam maps. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018.

[4] F. Boniardi, A. Valada, W. Burgard, and G. D. Tipaldi. Autonomous indoor robot navigation using a sketch interface for drawing maps and routes. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 2016.

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

[6] Y. Chebotar, Q. Vuong, K. Hausman, F. Xia, Y. Lu, A. Irpan, A. Kumar, T. Yu, A. Herzog, K. Pertsch, et al. Q-transformer: Scalable offline reinforcement learning via autoregressive q-functions. In *Conference on Robot Learning*, 2023.

[7] A. S. Chen, A. M. Lessing, A. Tang, G. Chada, L. Smith, S. Levine, and C. Finn. Commonsense reasoning for legged robot adaptation with vision-language models. *arXiv preprint arXiv:2407.02666*, 2024.

[8] H. Chen, H. Tan, A. Kuntz, M. Bansal, and R. Alterovitz. Enabling robots to understand incomplete natural language instructions using commonsense reasoning. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020.

[9] J. Christian Andersen, O. Ravn, and N. A. Andersen. Autonomous rule-based robot navigation in orchards. *IFAC Proceedings Volumes*, 2010.

[10] R. Dadashi, L. Hussenot, D. Vincent, S. Girgin, A. Raichuk, M. Geist, and O. Pietquin. Continuous control with action quantization from demonstrations. *arXiv preprint arXiv:2110.10149*, 2021.

[11] Y. Ding, C. Florensa, M. Phielipp, and P. Abbeel. Goal-conditioned imitation learning. *Advances in Neural Information Processing Systems*, 2019.

[12] Y. Du, N. J. Hetherington, C. L. Oon, W. P. Chan, C. P. Quintero, E. Croft, and H. Machiel Van der Loos. Group surfing: A pedestrian-based approach to sidewalk robot navigation. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 6518–6524, 2019. doi: 10.1109/ICRA.2019.8793608.

[13] Epic Games. Unreal engine. `https://www.unrealengine.com`, [Accessed: 2024.09.15].

[14] J. Gu, E. Stefani, Q. Wu, J. Thomason, and X. Wang. Vision-and-language navigation: A survey of tasks, methods, and future directions. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2022.

[15] J. Gu, S. Kirmani, P. Wohlhart, Y. Lu, M. G. Arenas, K. Rao, W. Yu, C. Fu, K. Gopalakrishnan, Z. Xu, P. Sundaresan, P. Xu, H. Su, K. Hausman, C. Finn, Q. Vuong, and T. Xiao. Rt-trajectory: Robotic task generalization via hindsight trajectory sketches. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*, 2024.

[16] F. Gul, W. Rahiman, and S. S. Nazli Alhady. A comprehensive study for robot navigation techniques. *Cogent Engineering*, 2019.

[17] N. Hirose, F. Xia, R. Martín-Martín, A. Sadeghian, and S. Savarese. Deep visual mpc-policy learning for navigation. *IEEE Robotics and Automation Letters*, 2019.

[18] C. Huang, O. Mees, A. Zeng, and W. Burgard. Visual language maps for robot navigation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023.

[19] E. Jang, A. Irpan, M. Khansari, D. Kappler, F. Ebert, C. Lynch, S. Levine, and C. Finn. BC-z: Zero-shot task generalization with robotic imitation learning. In *5th Annual Conference on Robot Learning*, 2021.

[20] H. Karnan, G. Warnell, X. Xiao, and P. Stone. Voila: Visual-observation-only imitation learning for autonomous navigation. In *2022 International Conference on Robotics and Automation (ICRA)*, 2022.

[21] H. Laurençon, L. Tronchon, M. Cord, and V. Sanh. What matters when building vision-language models? *arXiv preprint arXiv:2405.02246*, 2024.

[22] S. Lee, Y. Wang, H. Etukuru, H. J. Kim, N. M. M. Shafiullah, and L. Pinto. Behavior generation with latent actions. In *Proceedings of the 41st International Conference on Machine Learning*, 2024.

[23] Z. Li, K. Li, S. Wang, S. Lan, Z. Yu, Y. Ji, Z. Li, Z. Zhu, J. Kautz, Z. Wu, et al. Hydra-mdp: End-to-end multimodal planning with multi-target hydra-distillation. *arXiv preprint arXiv:2406.06978*, 2024.

[24] Z. Li, X. Liu, H. Wang, J. Song, F. Xie, and K. Wang. Research on robot path planning based on point cloud map in orchard environment. *IEEE Access*, 2024.

[25] B. Liu, Z. Guan, B. Li, G. Wen, and Y. Zhao. Research on slam algorithm and navigation of mobile robot based on ros. In *2021 IEEE International Conference on Mechatronics and Automation (ICMA)*, 2021.

[26] L. Metz, J. Ibarz, N. Jaitly, and J. Davidson. Discrete sequential prediction of continuous actions for deep rl. *arXiv preprint arXiv:1705.05035*, 2017.

[27] NVIDIA. Isaac Sim - Robotics Simulation and Synthetic Data — NVIDIA Developer. `https://developer.nvidia.com/isaac/sim`, Accessed: 2024.09.15.

[28] Open Robotics. Github - ros-planning/navigation, 2023. `https://github.com/ros-planning/navigation` version: noetic, 1.17.3 accessed: 2024.09.15.

[29] X. Pan, T. Zhang, B. Ichter, A. Faust, J. Tan, and S. Ha. Zero-shot imitation learning from demonstrations for legged robot visual navigation. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020.

[30] N. M. Shafiullah, Z. Cui, A. A. Altanzaya, and L. Pinto. Behavior transformers: Cloning k modes with one stone. *Advances in neural information processing systems*, 2022.

[31] D. Shah, A. Sridhar, A. Bhorkar, N. Hirose, and S. Levine. Gnm: A general navigation model to drive any robot. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023.

[32] D. Shah, A. Sridhar, N. Dashora, K. Stachowicz, K. Black, N. Hirose, and S. Levine. Vint: A foundation model for visual navigation. In *Proceedings of The 7th Conference on Robot Learning*, 2023.

[33] M. Skubic, D. Anderson, S. Blisard, D. Perzanowski, and A. Schultz. Using a hand-drawn sketch to control a team of robots. *Autonomous Robots*, 2007.

[34] A. Sridhar, D. Shah, C. Glossop, and S. Levine. Nomad: Goal masked diffusion policies for navigation and exploration. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024.

[35] H. Taheri and S. R. Hosseini. Deep reinforcement learning with enhanced ppo for safe mobile robot navigation. *arXiv preprint arXiv:2405.16266*, 2024.

[36] J.-P. Töberg, A.-C. Ngonga Ngomo, M. Beetz, and P. Cimiano. Commonsense knowledge in cognitive robotics: a systematic literature review. *Frontiers in Robotics and AI*, 2024.

[37] H. Wang, L. Zhang, Q. Kong, W. Zhu, J. Zheng, L. Zhuang, and X. Xu. Motion planning in complex urban environments: An industrial application on autonomous last-mile delivery vehicles. *Journal of Field Robotics*, 39(8):1258–1285, 2022.

[38] E. Wijmans, A. Kadian, A. Morcos, S. Lee, I. Essa, D. Parikh, M. Savva, and D. Batra. Dd-ppo: Learning near-perfect pointgoal navigators from 2.5 billion frames. In *International Conference on Learning Representations*, 2020.

[39] W. Wu, T. Chang, X. Li, Q. Yin, and Y. Hu. Vision-language navigation: a survey and taxonomy. *Neural Computing and Applications*, 2023.

[40] A. Yang, B. Yang, B. Hui, B. Zheng, B. Yu, C. Zhou, C. Li, C. Li, D. Liu, F. Huang, et al. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2024.

[41] M. Yin, T. Li, H. Lei, Y. Hu, S. Rangan, and Q. Zhu. Zero-shot wireless indoor navigation through physics-informed reinforcement learning. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024.

[42] M. Zare, P. M. Kebria, A. Khosravi, and S. Nahavandi. A survey of imitation learning: Algorithms, recent developments, and challenges. *IEEE Transactions on Cybernetics*, 2024.

[43] X. Zhai, B. Mustafa, A. Kolesnikov, and L. Beyer. Sigmoid loss for language image pre-training. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

[44] J. Zhang, K. Wang, R. Xu, G. Zhou, Y. Hong, X. Fang, Q. Wu, Z. Zhang, and W. He. Navid: Video-based vlm plans the next step for vision-and-language navigation. *arXiv preprint arXiv:2402.15852*, 2024.

[45] G. Zhou, Y. Hong, and Q. Wu. Navgpt: Explicit reasoning in vision-and-language navigation with large language models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 2024.

[46] K. Zhou, K. Zheng, C. Pryor, Y. Shen, H. Jin, L. Getoor, and X. E. Wang. Esc: Exploration with soft commonsense constraints for zero-shot object navigation. In *Proceedings of the 40th International Conference on Machine Learning*, 2023.

[47] W. Zu, W. Song, R. Chen, Z. Guo, F. Sun, Z. Tian, W. Pan, and J. Wang. Language and sketching: An llm-driven interactive multimodal multitask robot navigation framework. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024.