

VL-TGS: Trajectory Generation and Selection Using Vision Language Models in Mapless Outdoor Environments

Daeun Song , Member, IEEE, Jing Liang , Xuesu Xiao , and Dinesh Manocha , Fellow, IEEE

Abstract—We present a multi-modal trajectory generation and selection algorithm for real-world mapless outdoor navigation in human-centered environments. Such environments contain rich features like crosswalks, grass, and curbs, which are easily interpretable by humans, but not by mobile robots. We aim to compute suitable trajectories that (1) satisfy the environment-specific traversability constraints and (2) generate human-like paths while navigating on crosswalks, sidewalks, etc. Our formulation uses a Conditional Variational Autoencoder (CVAE) generative model enhanced with traversability constraints to generate multiple candidate trajectories for global navigation. We develop a visual prompting approach and leverage the Visual Language Model's (VLM) zero-shot ability of semantic understanding and logical reasoning to choose the best trajectory given the contextual information about the task. We evaluate our method in various outdoor scenes with wheeled robots and compare the performance with other global navigation algorithms. In practice, we observe an average improvement of 20.81% in satisfying traversability constraints and 28.51% in terms of human-like navigation in four different outdoor navigation scenarios.

Index Terms—Motion and path planning, task and motion planning, integrated planning and learning.

I. INTRODUCTION

MAPLESS outdoor navigation requires robots to compute trajectories or directions in large-scale environments without relying on pre-built maps. This problem is particularly

Received 4 December 2024; accepted 30 March 2025. Date of publication 10 April 2025; date of current version 29 April 2025. This article was recommended for publication by Associate Editor J. Stueckler and Editor S. Behnke upon evaluation of the reviewers' comments. This work was supported in part by ARO through University of Maryland at College Park under Grant W911NF2310046 and Grant W911NF2310352, in part by U.S. Army Cooperative under Grant W911NF2120076, in part by the National Science Foundation NSF through George Mason University under Grant 2350352, in part by Army Research Laboratory ARL under Grant W911NF2220242, Grant W911NF2320004, Grant W911NF2420027, and Grant W911NF2520011, in part by Air Force Research Laboratory (AFRL) and U.S. Air Forces Central AFCENT, under Grant GS00Q14OADU309, and in part by Google DeepMind (GDM), Clearpath Robotics, and Raytheon Technologies (RTX). (Daeun Song and Jing Liang contributed equally to this work.) (Corresponding author: Daeun Song.)

Daeun Song was with the University of Maryland, College Park, MD 20742 USA. He is now with the Department of Computer Science, George Mason University, Fairfax, VA 22030 USA (e-mail: dsong26@gmu.edu).

Xuesu Xiao is with the Department of Computer Science, George Mason University, Fairfax, VA 22030 USA (e-mail: xiao@gmu.edu).

Jing Liang and Dinesh Manocha are with the Department of Computer Science, University of Maryland, College Park, MD 20742 USA (e-mail: jingl@umd.edu; dmanocha@umd.edu).

This article has supplementary downloadable material available at <https://doi.org/10.1109/LRA.2025.3559822>, provided by the authors.

Digital Object Identifier 10.1109/LRA.2025.3559822

important for global navigation in outdoor settings, where creating and maintaining accurate maps is impractical due to dynamic changes such as constructions [1], [2]. Unlike map-based methods that depend on detailed geometric representations of the environment [3], [4], [5], mapless techniques rely directly on sensory input [6], [7], requiring robots to adapt to environmental changes and navigate through unknown spaces without the need for prior knowledge.

Traditionally, both map-based and mapless navigation approaches have relied on traversability analysis based on geometric shapes, often using LiDAR data to identify navigable regions [6], [7], [8]. While this approach is effective for detecting larger obstacles and general terrain features, it faces challenges in nuanced environments [9], [10]. Features such as short grass, curbs, and low-profile flower beds can be challenging for LiDAR to detect due to their subtle and low-profile characteristics. Additionally, while geometric environmental data is sufficient for navigation in obstacle-rich environments, it falls short in human-centered environments.

Navigating human-centered outdoor environments requires advanced scene understanding to ensure safety and reliability [11]. Robots must not only recognize physical features, such as walkways, crosswalks, and paved paths, but also interpret their intended use within the environment and navigate accordingly. For example, paved roadways may only be temporarily used when construction blocks the sidewalk, but they can always be used to cross a street when marked with a zebra crossing. This involves identifying areas designated for pedestrian movement, detecting obstacles or temporary changes, and understanding how these elements influence viable paths. Achieving this requires contextual reasoning to understand and adapt to the implicit rules and expectations of human-centered environments [12].

To build such contextual understanding of the environment, many existing methods [9], [13] rely on segmentation or classification [14], [15]. However, these require extensive training with ground truth data and are limited to labeled datasets. This limitation hinders their generalizability to unknown scenes. Recent advances in Large Language Models (LLMs) and Vision Language Models (VLMs) have demonstrated strong zero-shot capabilities across a wide range of tasks, including logical reasoning [16], [17] and visual understanding [18], [19]. VLMs, in particular, have the ability to process and understand both visual and textual information, enabling them to perform a wide range of multi-modal tasks. Their ability to reason contextually and adapt their outputs to align with implicit environmental rules makes them ideal for navigating human-centered outdoor spaces.

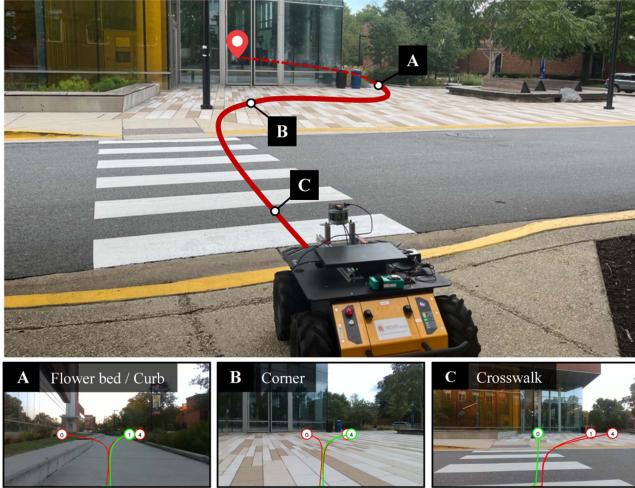


Fig. 1. Trajectories generated and selected using VL-TGS in outdoor navigation. The example path includes three different types of scenarios: (A) flower bed and curb, (B) corner, and (C) crosswalk. On the top, the map pin icon marks the goal behind the building, with the red solid or dashed line highlighting the robot's path. On the bottom, candidate trajectories are marked in red lines with numbers. The green path corresponds to the trajectory computed using VL-TGS. Overall, VL-TGS is capable of generating diverse, geometrically traversable paths and selecting semantically feasible trajectories for navigation in human-centered environments.

Main Results: We present VL-TGS, a novel multi-modal approach for trajectory generation and selection in mapless outdoor navigation (Fig. 1). Our method combines LiDAR-based geometric information with RGB image data for comprehensive traversability analysis and scene understanding. Using a CVAE-based [20] approach, we first generate multiple candidate trajectories based on the LiDAR scene perception. A VLM is then employed for trajectory selection based on the environmental context understanding through RGB image data. While VLMs lack the capability to produce precise spatial outputs, they can effectively utilize visual annotations to guide the selection process among a discrete set of coarse options [21], [22], [23]. By incorporating VLMs, our approach enables human-like decision-making to select optimal trajectories from the candidates, ensuring they align with geometric traversability constraints while addressing the contextual demands of global navigation. We demonstrate the effectiveness of our approach in outdoor scenarios featuring diverse human-centered environments and navigation challenges, such as crossing streets at crosswalks and adhering to walkways. The major contributions of our work include:

- 1) A novel integrated trajectory generation and selection method, VL-TGS, to generate multiple candidate trajectories using a CVAE-based approach and to select the most suitable trajectory using the VLM with visual prompting. Our CVAE-based trajectory generation method generates multiple candidate trajectories that are traversable considering the geometrical information retrieved from the LiDAR sensor. Our VLM-based trajectory selection method selects the best trajectory, which is traversable, in terms of both a geometric and semantic manner suitable for a human-centered environment.
- 2) We explore the use of a visual prompting approach to enhance the spatial reasoning capabilities of VLMs in

the context of trajectory selection. By incorporating visual markers such as lines and numerical indicators within the RGB image, we provide explicit guidance to the VLM. We conduct ablation studies, first demonstrating the importance of providing high-quality candidate trajectories, and then comparing the effectiveness of having a visual marking method.

- 3) We evaluate VL-TGS in four different outdoor scenarios. We measure the satisfaction rate of traversability constraints and the Fréchet distance with respect to a human-teleoperated trajectory. We compare the results with state-of-the-art trajectory generation approaches. We observe an average improvement of 20.81% in the traversability satisfaction rate and 28.51% in the Fréchet distance. We also qualitatively demonstrate the benefits of our approach over other methods.

II. RELATED WORK

This section reviews related works on outdoor robot navigation, with a particular focus on trajectory generation.

A. Outdoor Robot Navigation

Reinforcement-learning-based motion planning approaches [24], [25] use an end-to-end structure to take observations and generate actions or trajectories. However, these methods are designed for short-range navigation, and on-policy reinforcement learning approaches also suffer from the reality gap. Map reconstruction with path planning approaches [26], [27] provides a solution for global planning by building a map during navigation, but these approaches require a large memory for the global map. To address this issue, NoMaD [28] and ViNT [29] use topological maps to reduce memory usage for navigation, but these approaches require topological nodes to be predefined, making them unsuitable for fully unknown environments. To overcome these limitations, our approach uses a CVAE-based trajectory generation method [6] to generate trajectories and leverages VLMs to select the optimal trajectory to reach the goal.

B. Vision Language Models in Navigation

Recent breakthroughs in Language Foundation Models (LFMs) [30], encompassing VLMs and LLMs, demonstrate significant potential for robotic navigation. LM-Nav [31] employs GPT-3 and CLIP [19] to extract landmark descriptions from text-based navigation instruction and ground them in images, effectively guiding a robot to the goal in outdoor environments. VLMaps [32] propose a spatial map representation by fusing vision-language features with a 3D map that enables natural language-guided navigation. CoW [33] performs zero-shot language-based object navigation by combining CLIP-based maps and traditional exploration methods. Most of these researches focus on utilizing VLMs for high-level navigation guidance by extracting text-image scene representation. For low-level navigation behaviors, VLM-Social-Nav [34] explores the ability of VLM to extract socially compliant navigation behavior with the interaction with social entities like humans. CoNVOI [35] uses visual annotation to extract a sequence of waypoints from camera observation to navigate robots. PIVOT [22] uses visual prompting and optimization with VLMs in various low-level robot control tasks

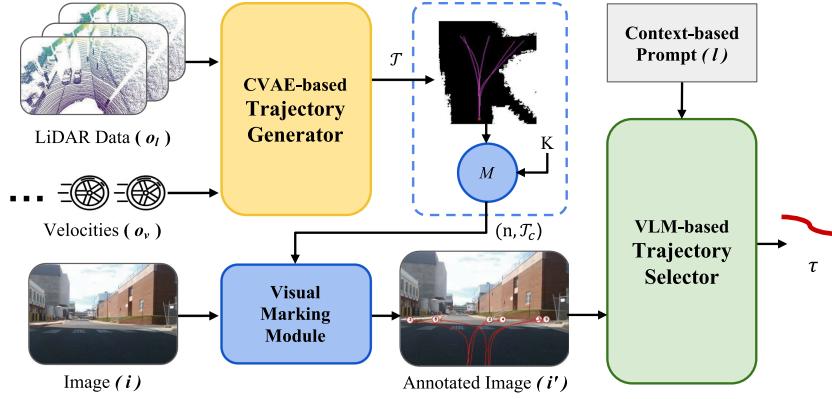


Fig. 2. **Architecture:** Our approach consists of two stages: CVAE-based trajectory generation and VLM-based trajectory selection. In the first stage, our attention-based CVAE takes consecutive frames of LiDAR point clouds and robot velocities as input, generating multiple diverse trajectories. These trajectories are sorted and visually marked with lines and numbers in the robot-view RGB image. In the second stage, our VLM-based trajectory selection module identifies the best trajectory number based on semantic feasibility, ensuring it lies on the sidewalk, avoids structures, crosses at zebra crossings, and adheres to other contextual rules.

including indoor navigation. It shows the potential of a visual prompting approach for VLMs in robotic and spatial reasoning domains.

Building on these approaches, our work uses a VLM to guide low-level navigation behavior by understanding contextual and semantic information about the surroundings. We use visual annotations [21], [22], [35], [36], such as lines and numbers, to aid the VLM to effectively comprehend spatial information. Instead of randomly generating the candidates like in PIVOT [22], we use a generative model-based trajectory generation approach to produce diverse candidate trajectories that ensure traversability for the VLM to choose from.

III. APPROACH

In this section, we formulate the problem of mapless global navigation and describe our approach.

A. Overview

Our approach computes a trajectory in a mapless environment for global navigation. Mapless global navigation requires a robot to reach a distant target beyond its immediate surroundings without relying on a pre-built map. To achieve this, we utilize multi-modal sensor data, combining both geometric and RGB visual information, to iteratively generate local trajectories that guide the robot towards the goal. Our approach follows a two-stage pipeline, as illustrated in Fig. 2. In the first stage, we generate multiple candidate trajectories, each spanning a fixed length (e.g., 10 m) that satisfy the geometric traversability constraints. Then, we select the best trajectory based on human-like decision-making. Given a target goal $g \in \mathcal{O}_g$, we use a GPS sensor to provide the relative position between the target and the current location. Our goal is to compute a trajectory, τ , that aims to provide the best path to the goal, and that satisfies the traversability constraints of the scenario, $\tau = VL\text{-TGS}(\ell, \mathbf{i}, \mathbf{o}, g)$, where $\mathbf{o} = \{\mathbf{o}_l, \mathbf{o}_v, \mathbf{i}\}$ represents the robot's observations. $\mathbf{o}_l \in \mathcal{O}_l$ represents LiDAR observations, $\mathbf{o}_v \in \mathcal{O}_v$ indicates the robot's velocity, and $\mathbf{i} \in \mathcal{I}$ represents the RGB images from the camera. $\ell \in \mathcal{L}$ represents the language instructions to the Vision-Language Models (VLMs) for acquiring traversable trajectories.

We use Conditional Variational Autoencoder (CVAE) [6] to process the geometric information, $\mathbf{o}_l \in \mathcal{O}_l$, from the LiDAR sensor and the consecutive velocities, $\mathbf{o}_v \in \mathcal{O}_v$, from the robot's odometer. We efficiently generate a set of trajectories lying in geometrically traversable areas, $\mathcal{T} = \text{CVAE}(\mathbf{o}_l, \mathbf{o}_v)$. These generated trajectories cannot handle geometrically similar but color-semantically different situations, such as crosswalks as shown in Fig. 1(C). Therefore, we use VLMs to provide scene understanding from the RGB images.

However, the generated real-world waypoints from CVAE and the image observations are in two different modalities. To fuse these, we overlay the trajectories onto the images. VLMs are then used to assess whether the trajectories align with the contextual constraints of the environment. We assume that VLMs can infer common-sense reasoning from the images. We place these numbers at the end of each trajectory, starting from 0. The numbers indicate the order of distances to the goal, with the lowest number corresponding to the trajectory with the shortest distance. Thus, we map the real-world trajectories to image pixel-level objects by

$$(\mathbf{n}, \mathcal{T}_c) = M(\text{CVAE}(\mathbf{o}_l, \mathbf{o}_v), K), \quad (1)$$

where K denotes the conversion matrix from the real-world LiDAR frame to the image plane, \mathcal{T}_c denotes the converted trajectories, and $\mathbf{n} \in \mathcal{N}$ are the numbers corresponding to each trajectory.

Given the language instruction ℓ , the image \mathbf{i} with the converted trajectories \mathcal{T}_c , and numbers $\mathbf{n} \in \mathcal{N}$, our VLM selects one traversable trajectory based on the color-semantic understanding of the scenarios:

$$\tau = \text{VLM}(\ell, \mathbf{i}, \mathcal{T}_c, \mathbf{n}). \quad (2)$$

We choose the trajectory with the highest probability as the human-like trajectories, $\max P(\tau | \ell, \mathbf{i}, \mathcal{T}_c, \mathbf{n})$. Therefore, the problem is defined as:

$$\max P(\tau | \ell, \mathbf{i}, \mathcal{T}_c, \mathbf{n}). \quad (3)$$

B. Geometry-Based Trajectory Generation

The trajectory set, \mathcal{T} , is generated by a CVAE to generate trajectories with associated confidences. For each observation

Algorithm 1: Multi-Modal Trajectory Generation and Selection Algorithm.

```

Given : LiDAR point cloud  $\mathbf{o}_l$ , robot's velocities  $\mathbf{o}_v$ ,  

         transformation matrix  $K$ , threshold  $d_t$ ,  

         instruction  $\ell$ , RGB image  $\mathbf{i}$ 
Initialize: trajectory set  $\mathcal{T} = \{\}$ , time stamp  $t = 2$ 
1 while the robot is running do
2    $\mathcal{T}_n = \text{CVAE}(\mathbf{o}_l, \mathbf{o}_v);$ 
3    $\mathcal{T} = \mathcal{T}_n \cup \mathcal{T};$ 
4    $(\mathbf{n}, \mathcal{T}_c) = M(\mathcal{T}, K);$ 
5    $\tau = \text{VLM}(\ell, \mathbf{i}, \mathcal{T}_c, \mathbf{n});$ 
6   if  $t_{\mathcal{T}} > t$  then
7     |  $\mathcal{T}.\text{DEQUEUE}();$ 
8 end

```

$\{\mathbf{o}_l, \mathbf{o}_v\}$, we calculate the condition value $\mathbf{c} = f_e(\mathbf{o}_l, \mathbf{o}_v)$ for the CVAE decoder, where $f_e(\cdot)$ denotes the perception encoder. The embedding vector is then calculated from \mathbf{c} as $\mathbf{z} = f_z(\mathbf{c})$, with $f_z(\cdot)$ representing a neural network.

To generate a sufficient number of candidates for the robot's navigation, we need to create multiple diverse trajectories that cover all traversable areas in front of the robot. Since the decoder is designed to generate a single trajectory from one embedding vector, producing a variety of diverse trajectories requires the use of representative and varied embedding vectors. We project the embedding vector \mathbf{z} onto orthogonal axes by linear transformations, each projected vector corresponding to one traversable area. Then we generate trajectories based on the condition \mathbf{c} :

$$\mathbf{z}_k = A_k(\mathbf{c})\mathbf{z} + b_k(\mathbf{c}) = h_{\psi_k}(\mathbf{z}), \quad (4)$$

where h_{ψ_k} denotes the linear transformation of \mathbf{z} . Using each embedding vector \mathbf{z}_k , the decoder generates a trajectory τ_k , as $p(\tau_k | \mathbf{z}_k, \mathbf{c}, \tilde{\mathcal{T}}_k)$. $\tau_k \in \mathcal{T}$ represents generated trajectories. \mathbf{z}_k and $\tilde{\mathcal{T}}_k$ are the embedding vectors of the current trajectory and the set of other trajectory embeddings, respectively. The training of the trajectory generator is the same as MTG [6], where we use traversability loss, CVAE lower bound, and diversity loss to train the model.

C. VLM-Based Trajectory Selection

Algorithm 1 highlights our procedure of using VLMs to select a suitable trajectory from candidate trajectories. $t_{\mathcal{T}}$ denotes the time steps \mathcal{T} contains. \mathcal{T}_n denotes a new set of trajectories generated by CVAE. While the generated trajectories \mathcal{T}_n effectively cover the traversable areas in front of the robot [6], the deep-learning-based generative model cannot guarantee the consistent generation of traversable trajectories. To address this, we sample consecutive $t = 2$ time steps, introducing redundancy to increase the likelihood that at least one of the generated trajectories will be traversable. Given the collected trajectories in \mathcal{T} , we convert them to the image plane with numbers, where we sort the trajectories in terms of heuristic, which is the distance between the last waypoint of the trajectory and the goal, as shown in (1).

Considering that trajectories generated at consecutive time steps often overlap significantly, we refine the set of candidates \mathcal{T} . This is done by selecting only representative trajectories to

form a subset $\mathcal{T}' \subseteq \mathcal{T}$ based on their Hausdorff distances:

$$\forall \tau_n, \tau_m \in \mathcal{T}', d_h(\tau_n, \tau_m) > d_t, \text{ where } n \neq m, \quad (5)$$

where $d_h(\cdot, \cdot)$ represents the Hausdorff distance. This process removes trajectories that are too similar, improving the clarity of visual annotations on the image while ensuring diversity.

We then project the trajectories \mathcal{T}' from the robot's frame to the image plane by transformation matrices K , $\mathcal{T}_c = P_c(\mathcal{T}', K)$. Following the trajectory generation sequence, we annotate the trajectories with numbers, \mathbf{n} .

Finally, we use the VLM to select the best trajectory in terms of satisfying traversability and social compliance. The annotated trajectories $(\mathbf{n}, \mathcal{T}_c)$ and the current observation image \mathbf{i} are input into the VLM with the prompt instruction ℓ . The VLM selects the best trajectory, τ , in terms of traversability, social compliance, and traveling distance to the goal, as shown in (2). The following is the instruction prompt instruction ℓ provided to the model:

I am a wheeled robot that cannot go over high bumps. This is the image I am seeing right now.

Pick one path that I should follow to navigate safely towards the goal, like what humans do. Remember that I must walk on pavements, avoid rough, bumpy terrains, and follow the rules. I cannot go over/under the curbs. The lowest number indicates the shortest path to the goal. Pick only one.

Provide the answer in this form: { 'trajectory': [] }

Given the selected trajectory τ , our motion planner generates the corresponding robot action \mathbf{a} to follow it. The VLM is re-prompted each time it returns a response. Although our VLM-based trajectory selector operates at a relatively low frequency, i.e., every 2 to 4 seconds, the trajectory generator efficiently produces 10 m trajectories, ensuring the latency remains manageable.

IV. EXPERIMENTAL RESULTS

In this section, we present the implementation details and the experimental results of our approach.

A. Implementation Details

Our approach is tested on a Clearpath Husky equipped with a Velodyne VLP16 LiDAR, a Realsense D435i camera, and a laptop with an Intel i7 CPU and an Nvidia GeForce RTX 2080 GPU. We use CVAE with an attention mechanism to generate multiple trajectories [6] (approximately 10 m each) and use GPT-4V [37] to select the best traversable trajectory.

The training dataset [38] for our CVAE-based trajectory generation model contains three parts: 1) LiDAR point cloud and robot velocities, 2) binary traversability maps, shown in the right column of Figs. 4 and 5, 3) randomly generated diverse targets with the shortest ground truth trajectories to the targets. The binary traversability map is constructed from LiDAR points and is used only for training and evaluation. The map is not used during inference.

To validate VL-TGS, we present qualitative and quantitative results compared with MTG [6], ViNT [29], NoMaD [28], PIVOT [22], and CoNVOI [35]. We evaluate the performance in four challenging benchmark scenarios:

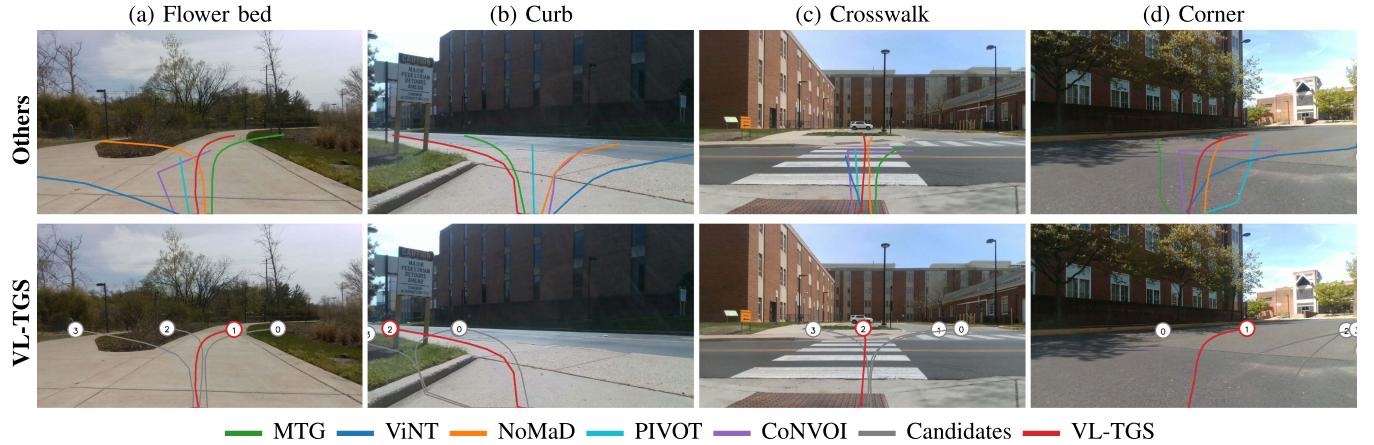


Fig. 3. **Qualitative Results:** The top row shows the generated trajectories using all the methods, MTG [6] in green, ViNT [29] in blue, NoMaD [28] in orange, PIVOT [22] in cyan, CoNVOI [35] in purple, and VL-TGS in red. The bottom row shows the candidate trajectories in gray marked with numbers and the selected trajectory in red using VL-TGS. VL-TGS can generate and select a trajectory that is both geometrically and semantically feasible.

- **Flower bed:** A robot navigating a paved area next to a flower bed. The robot must stay on the paved path and avoid entering the flower bed.
- **Curb:** A robot navigating on a sidewalk, which is distinctly separated from the roadway by a curb. The robot must stay on a sidewalk or select a traversable trajectory to go around the curb.
- **Crosswalk:** A robot crossing the street. The robot must stay on the crosswalk when crossing the street.
- **Behind the corner:** When the target is behind an obstruction, and there is a large open space ahead, the straight path may lead to an obstacle. The robot must choose a trajectory to navigate around the corner.

These scenarios pose challenges for navigation without semantic understanding, yet they are common in human-centered environments.

B. Qualitative Results

Fig. 3 shows the resulting robot trajectories corresponding to six different approaches in four different scenarios. The upper row shows the trajectories generated by all the comparison methods including ours and the lower row shows the results of VL-TGS with the candidate trajectories (gray) and the selected one (red).

As MTG relies solely on LiDAR’s geometric data, it is unable to deal with traversability differences in flower beds, curbs, and crosswalks, where structure alone provides little distinction. Also, in the corner case where the goal is located around a bend or behind a structure, MTG tends to fail by attempting to cut through rather than effectively navigating around the structure. The performances of ViNT and NoMaD heavily depend on the quality of pre-built topological maps. While they perform well when following straight paths with distinct visual features, such as a crosswalk, they often struggle in environments with turns or significant scene variations. While PIVOT selects the most semantically feasible trajectory from the given candidates, it does not explicitly detect geometric information and its random trajectory generation disregards both geometric and semantic information, potentially resulting in no viable options for the VLM to choose from. Compared to other methods, CoNVOI

generally produces trajectories that are both geometrically and semantically feasible. However, its zigzag motion results in non-smooth robot movements. As shown in the bottom row of each scenario in Fig. 3, our approach produces diverse trajectories and selects the best one that is traversable and contextually appropriate.

C. Quantitative Results

To further validate VL-TGS, we evaluate the methods using two different metrics:

- **Traversability:** The ratio of the generated trajectory lying on a traversable area. The binary traversability map, initially generated using LiDAR, is used for evaluation. This metric is calculated as:

$$tr(\mathcal{A}, \hat{\tau}) = \sum_{m=1}^M c(\mathcal{A}, \mathbf{w}_m), \quad \mathbf{w}_m \in \hat{\tau}, \quad (6)$$

where $c(\cdot, \cdot)$ tells if the waypoint \mathbf{w}_m is in the traversable area \mathcal{A} .

- **Fréchet Distance w.r.t. Human Tele-operation:** Fréchet Distance [39] is one of the measures of similarity between two curves. We measure the similarity between the trajectories generated by the methods and the trajectory tele-operated by human.

A lower distance indicates a higher degree of similarity.

Table I reports the results averaged over 20 different frames, with five repetitions for each frame, scenario, and method. Fig. 3 shows one of the examples. In the Input column, L indicates LiDAR point cloud and I indicates RGB images. While MTG, ViNT, NoMad, and PIVOT rely on a single sensory input, CoNVOI and VL-TGS utilize both LiDAR point clouds and RGB images. The results demonstrate that VL-TGS outperforms other state-of-the-art approaches in most of the cases. Specifically, we achieve at least 3.35% and at most 47.74% improvement in terms of average traversability, and at least 19.62% and at most 40.99% improvement in terms of average Fréchet distance. Overall, the average improvement rates are approximately 20.81% for traversability and 28.51% in Fréchet distance.

TABLE I
QUANTITATIVE RESULTS: COMPARISONS WITH STATE-OF-ART METHODS

Metric	Method	Input	Scenario			
			Flower bed	Curb	Crosswalk	Corner
Traversability (%) ↑	MTG [6]	L	58.19 ± 16.65	67.12 ± 15.65	61.82 ± 10.95	44.71 ± 18.35
	ViNT [29]	I	63.62 ± 18.49	78.37 ± 17.94	84.78 ± 3.16	44.95 ± 17.16
	NoMaD [28]	I	75.64 ± 15.04	83.13 ± 10.04	79.24 ± 13.36	77.38 ± 18.59
	PIVOT [22]	I	64.75 ± 19.63	79.58 ± 12.86	76.78 ± 10.41	68.66 ± 15.76
	CoNVOI [35]	I+L	81.10 ± 9.98	75.68 ± 12.86	86.24 ± 12.63	88.46 ± 11.45
	VL-TGS (Ours)	I+L	87.22 ± 10.27	89.93 ± 7.11	87.44 ± 9.78	78.00 ± 7.79
Fréchet Distance (m) ↓	MTG [6]	L	6.61 ± 1.91	8.40 ± 6.30	10.42 ± 2.53	9.93 ± 3.04
	ViNT [29]	I	10.43 ± 2.92	10.78 ± 3.08	8.94 ± 2.29	12.71 ± 2.43
	NoMaD [28]	I	7.65 ± 3.32	8.71 ± 3.53	11.87 ± 2.99	9.62 ± 2.60
	PIVOT [22]	I	8.41 ± 1.85	7.86 ± 1.55	10.53 ± 3.00	9.48 ± 3.15
	CoNVOI [35]	I+L	11.64 ± 0.47	12.24 ± 1.12	11.33 ± 1.26	12.36 ± 2.15
	VL-TGS (Ours)	I+L	5.27 ± 1.65	7.93 ± 1.28	6.38 ± 2.95	8.49 ± 2.29

The bold values indicate the best-performing metrics for each scenario.

In Table I, we observe that MTG produces very low results in terms of traversability. This is not only because our benchmark scenarios were selected based on scenarios that are difficult to detect with LiDAR, but also because MTG often fails to consider traversability while focusing on optimality to the goal. In terms of Fréchet distance, MTG and VL-TGS produce good results because they output smooth trajectories similar to a human-operated trajectory we compare against. In contrast, CoNVOI generates a linear trajectory that differs significantly from typical human-operated trajectories, resulting in a lower similarity. CoNVOI generates short trajectories using only two waypoints, reducing the likelihood of waypoints landing in non-traversable areas and leading to a high traversability result. However, in practice, intermediate points may still fall into non-traversable regions. Both ViNT and NoMaD are image-based navigation approaches, but NoMaD outperforms ViNT in 3 out of 4 cases in terms of traversability and Fréchet distance. While both perform well in straight-line following scenarios (e.g., crosswalks), they tend to go off-course when robots are taking turns or the scenarios are dynamic. Additionally, since some of our flower bed and curb scenarios included smooth turns, their variance is notably high. As PIVOT generates random straight-line candidates, its performance is inconsistent, exhibiting high variation in results. The result demonstrates that VL-TGS generates human-like trajectories in human-centered environments while ensuring good traversability.

D. Ablation Studies

To evaluate the capability of different components of our innovations, we compare VL-TGS with two different settings. First, we compare by removing our CVAE-based trajectory generator. Instead, we randomly generate trajectories. This approach aligns with the method utilized by PIVOT, but we omit their iterative questioning mechanism as part of our ablation study. Second, we compare by removing our VLM-based trajectory selector. Instead, we select a trajectory by using a heuristic to select the shortest travel distance to the goal, which aligns with the approach, MTG.

Ablation on Trajectory Generator: Fig. 4 illustrates the ablation study to evaluate the effectiveness of our CVAE-based trajectory generator. The red lines and numbers are the inputs given to the VLM. The green line indicates the selected trajectory by the VLM. PIVOT randomly generates the sub-goal targets

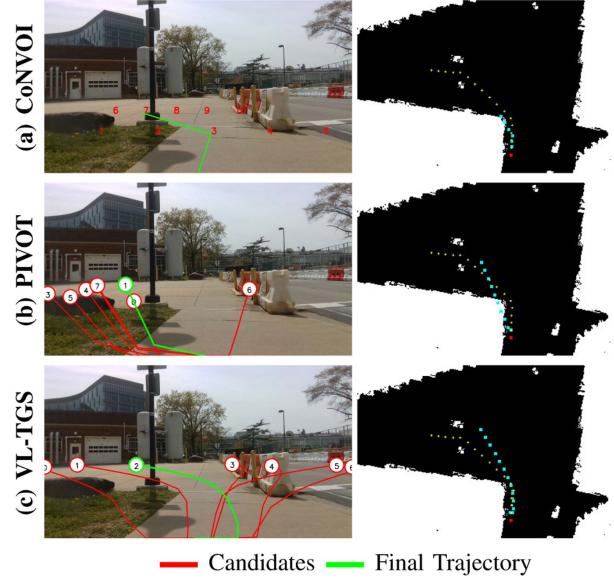


Fig. 4. **Ablation Study on the Trajectory Generator:** The left shows the generated candidate trajectories (red) and the selected trajectory (green) in the robot-view image. The right shows the top-down view image of the traversability map. The cyan color represents the final selected trajectory, and the yellow color represents the human-driven trajectory. Compared with CoNVOI [35] and PIVOT [22], VL-TGS generates the trajectory closest to the human-driven one, which keeps the robot on a safe pavement surface.

and linearly connects them. We randomly generate 10 endpoints that are within 5 m to 15 m ahead and then linearly connect the points to generate trajectories. It represents the approach of a VLM-based trajectory selector without a CVAE-based trajectory generator. Because the target is randomly generated, it often fails to generate good candidate trajectories. We also compare with CoNVOI, which adopts a different approach to generating candidates for the VLM. CoNVOI marks obstacle-free regions with numerical labels, employs the VLM to select suitable labels, and connects them with straight lines to form a trajectory. However, while the marked regions are obstacle-free, this method does not consider the waypoints between the labels. Consequently, the generated trajectories may intersect obstacles, as demonstrated in Fig. 4(b). VL-TGS utilizes the strengths of the CVAE-based

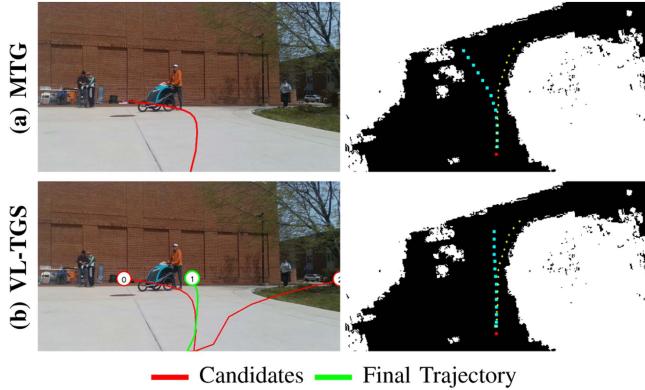


Fig. 5. **Ablation Study on the Trajectory Selector:** Compared with MTG [6], which selects trajectories based on the shortest distance heuristic, VL-TGS selects the trajectory closer to human-like decision-making, going around the large obstruction.

TABLE II
REAL WORLD EXPERIMENT RESULTS

Method	Number of Failures ↓	Travel Distance (m) ↓	Travel Time (sec.) ↓
MTG [6]	8	111.65	201
NoMaD [28]	6	100.79	168
CoNVOI [35]	13	115.77	257
VL-TGS (Ours)	4	97.53	151

trajectory generator to produce high-quality candidates for the VLM to evaluate and select from. The study highlights the critical role of having high-quality candidate trajectories, emphasizing the significance of an effective trajectory generator.

Ablation on Trajectory Selector: Fig. 5 illustrates the ablation study to evaluate the effectiveness of our VLM-based trajectory selector. MTG uses a CVAE-based approach to generate multiple trajectories and select the optimized trajectory based on a heuristic, the distance to the goal. It represents a CVAE-based trajectory generator without a VLM-based trajectory selector. When comparing MTG and VL-TGS, MTG generates a traversable trajectory but often overlooks small, dynamic obstacles such as humans. Additionally, when the target goal is located behind a building, MTG attempts to cut through the building, generating the shortest trajectory to the goal, whereas VL-TGS selects a trajectory that appropriately navigates around it. The study highlights the importance of the trajectory selector. Rather than relying on a heuristic to choose from candidate trajectories, our VLM-based trajectory selector enables human-like decision-making, driven by the robot's visual perception of the environment.

E. Real Robot Experiment

In order to demonstrate our approach in the real world, we performed experiments in the real world. Fig. 1 and Table II show the result of our robot experiments, showcasing a navigation task that incorporates all four scenarios. Each method was evaluated in a single run, and failures were counted whenever human

intervention was required to recover the robot. The supplementary video further highlights the resulting robot motions and compares them with other methods.

In our real robot experiments, we use GPS data to determine both the current robot position and the target goal, which is located approximately 100 m away, positioned behind a building obstruction. To navigate toward the goal, our approach continuously generates 10 m trajectories in a recursive manner. These generated trajectories are then executed using the Dynamic Window Approach (DWA) [40], ensuring smooth and adaptive motion planning. We compare our method against three alternative approaches: MTG, NoMaD, and CoNVOI. VL-TGS exhibits the least number of failures while achieving the shortest travel distance and time.

F. Discussions

Low Frequency of Online Large VLMs: A notable limitation of using large VLMs for navigation is their relatively low operational frequency, with outputs typically taking 2 to 4 seconds in our experiments. This latency makes them unsuitable for high-frequency real-time decision-making. However, our approach mitigates this issue effectively by generating relatively long trajectories of approximately 10 m, reducing the need for frequent updates. This allows the robot to continue moving smoothly without waiting for constant re-evaluation. Additionally, the motion planner ensures the robot continues to follow the selected trajectory while waiting for the VLM's decision. This design allows us to leverage the VLM's contextual reasoning capabilities without compromising navigation reliability. Furthermore, the improvements in VLM processing speed could further enhance system responsiveness.

More Challenging Scenarios: Although our benchmark scenarios focus on stationary environments, our approach is capable of handling dynamic scenarios involving moving obstacles. As illustrated in Fig. 5, our CVAE-based trajectory generator produces traversable candidate trajectories when evaluated against a traversability map. However, it often overlooks small, dynamic obstacles, such as humans. To complement this, our VLM-based trajectory selector incorporates such factors to identify and select feasible trajectories. Furthermore, while the VL-TGS module generates and selects a trajectory, the underlying motion planner ensures that the robot adheres to it while dynamically adjusting its motion in response to unexpected obstacles in real time. We employ the DWA as the motion planner for our experiments, but this can be replaced with any other local planning algorithm. In this letter, our primary focus is to demonstrate that VLM can effectively handle human-centered environments that require contextual understanding, such as pedestrian walkways and crossings, ensuring that navigation decisions align with social and environmental cues. We establish our benchmark to reflect these challenges.

V. CONCLUSION, LIMITATIONS, AND FUTURE WORK

We propose VL-TGS, a novel multi-modal Trajectory Generation and Selection approach for mapless outdoor navigation. VL-TGS integrates a CVAE-based trajectory generation method with a VLM-based trajectory selection process to compute geometrically and semantically feasible, human-like trajectories in human-centered outdoor environments. Our approach achieves a 20.81% improvement in traversability and a 28.51% improvement in similarity to human-operated trajectories on average.

Our method has a few limitations. Since VL-TGS relies on VLM, its performance can depend on the robustness of the VLM. However, with the ongoing improvements in VLM technology, it is expected that the robustness of our approach will also improve. Furthermore, our trajectory generation method can be substituted with more advanced approaches in the future, offering the potential for further performance enhancements.

REFERENCES

- [1] L. Wijayathunga, A. Rassau, and D. Chai, "Challenges and solutions for autonomous ground robot scene understanding and navigation in unstructured outdoor environments: A review," *Appl. Sci.*, vol. 13, no. 17, 2023, Art. no. 9877.
- [2] I. Jeong, Y. Jang, J. Park, and Y. K. Cho, "Motion planning of mobile robots for autonomous navigation on uneven ground surfaces," *J. Comput. Civil Eng.*, vol. 35, no. 3, 2021, Art. no. 04021001.
- [3] S. Ganesan, S. K. Natarajan, and J. Srinivasan, "A global path planning algorithm for mobile robot in cluttered environments with an improved initial cost solution and convergence rate," *Arabian J. Sci. Eng.*, vol. 47, no. 3, pp. 3633–3647, 2022.
- [4] P. Gao, Z. Liu, Z. Wu, and D. Wang, "A global path planning algorithm for robots using reinforcement learning," in *Proc. IEEE Int. Conf. Robot. Biomimetics*, 2019, pp. 1693–1698.
- [5] M. Psotka et al., "Global path planning method based on a modification of the wavefront algorithm for ground mobile robots," *Robotics*, vol. 12, no. 1, 2023, Art. no. 25.
- [6] J. Liang et al., "MTG: Mapless trajectory generator with traversability coverage for outdoor navigation," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2024, pp. 2396–2402.
- [7] J. Liang, A. Payandeh, D. Song, X. Xiao, and D. Manocha, "DTG: Diffusion-based trajectory generation for mapless global navigation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2024, pp. 5340–5347.
- [8] B. Suger, B. Steder, and W. Burgard, "Traversability analysis for mobile robots in outdoor environments: A semi-supervised learning approach based on 3D-LiDAR data," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2015, pp. 3941–3946.
- [9] K. Weerakoon, A. J. Sathyamoorthy, J. Liang, T. Guan, U. Patel, and D. Manocha, "GrASPE: Graph based multimodal fusion for robot navigation in outdoor environments," *IEEE Robot. Automat. Lett.*, vol. 8, no. 12, pp. 8090–8097, Dec. 2023.
- [10] A. J. Sathyamoorthy, K. Weerakoon, M. Elnoor, M. Russell, J. Pusey, and D. Manocha, "MIM: Indoor and outdoor navigation in complex environments using multi-layer intensity maps," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2024, pp. 10917–10924.
- [11] R. Möller et al., "A survey on human-aware robot navigation," *Robot. Auton. Syst.*, vol. 145, 2021, Art. no. 103837.
- [12] K. Charalampous, I. Kostavelis, and A. Gasteratos, "Recent trends in social aware robot navigation: A survey," *Robot. Auton. Syst.*, vol. 93, pp. 85–104, 2017.
- [13] J. Zhang, K. Yang, A. Constantinescu, K. Peng, K. Müller, and R. Stiefelhagen, "Trans4Trans: Efficient transformer for transparent object and semantic scene segmentation in real-world navigation assistance," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 10, pp. 19173–19186, Oct. 2022.
- [14] A. Kirillov et al., "Segment anything," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2023, pp. 4015–4026.
- [15] P. V. K. Borges et al., "A survey on terrain traversability analysis for autonomous ground vehicles: Methods, sensors, and challenges," *Field Robot.*, vol. 2, no. 1, pp. 1567–1627, Jul. 2022.
- [16] J. Austin et al., "Program synthesis with large language models," 2021, *arXiv:2108.07732*.
- [17] H. Ha and S. Song, "Semantic abstraction: Open-world 3D scene understanding from 2D vision-language models," in *Proc. Conf. Robot Learn.*, 2022.
- [18] J. -B. Alayrac et al., "Flamingo: A visual language model for few-shot learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2022, pp. 23716–23736.
- [19] A. Radford et al., "Learning transferable visual models from natural language supervision," in *Proc. 38th Int. Conf. Mach. Learn.*, 2021, pp. 8748–8763.
- [20] K. Sohn, H. Lee, and X. Yan, "Learning structured output representation using deep conditional generative models," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 3483–3491.
- [21] A. Shtedritski, C. Rupprecht, and A. Vedaldi, "What does CLIP know about a red circle? Visual prompt engineering for VLMs," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2023, pp. 11987–11997.
- [22] S. Nasiriany et al., "PIVOT: Iterative visual prompting elicits actionable knowledge for VLMs," 2024, *arXiv:2402.07872*.
- [23] J. Yang et al., "Set-of-mark prompting unleashes extraordinary visual grounding in GPT-4V," 2023, *arXiv:2310.11441*.
- [24] J. Liang et al., "Crowd-steer: Realtime smooth and collision-free robot navigation in densely crowded scenarios trained using high-fidelity simulation," in *Proc. 29th Int. Conf. Int. Joint Conf. Artif. Intell.*, 2021, pp. 4221–4228.
- [25] J. Hao et al., "Exploration in deep reinforcement learning: From single-agent to multiagent domain," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 7, pp. 8762–8782, Jul. 2024.
- [26] L. Schmid, M. Pantic, R. Khanna, L. Ott, R. Siegwart, and J. Nieto, "An efficient sampling-based method for online informative path planning in unknown environments," *IEEE Robot. Automat. Lett.*, vol. 5, no. 2, pp. 1500–1507, Apr. 2020.
- [27] H. Zhai, M. Egerstedt, and H. Zhou, "Path exploration in unknown environments using Fokker-Planck equation on graph," *J. Intell. Robot. Syst.*, vol. 104, no. 4, 2022, Art. no. 71.
- [28] A. Sridhar, D. Shah, C. Glossop, and S. Levine, "NoMaD: Goal masked diffusion policies for navigation and exploration," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2024, pp. 63–70.
- [29] D. Shah et al., "ViINT: A foundation model for visual navigation," in *Proc. 7th Annu. Conf. Robot Learn.*, 2023, pp. 711–733.
- [30] R. Bommasani et al., "On the opportunities and risks of foundation models," 2022, *arXiv:2108.07258*.
- [31] D. Shah et al., "LM-Nav: Robotic navigation with large pre-trained models of language, vision, and action," in *Proc. 6th Conf. Robot Learn.*, 2023, pp. 492–504.
- [32] C. Huang, O. Mees, A. Zeng, and W. Burgard, "Visual language maps for robot navigation," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2023, pp. 10608–10615.
- [33] S. Y. Gadre, M. Wortsman, G. Ilharco, L. Schmidt, and S. Song, "CoWs on pasture: Baselines and benchmarks for language-driven zero-shot object navigation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 23171–23181.
- [34] D. Song, J. Liang, A. Payandeh, A. H. Raj, X. Xiao, and D. Manocha, "VLM-Social-Nav: Socially aware robot navigation through scoring using vision-language models," *IEEE Robot. Automat. Lett.*, vol. 10, no. 1, pp. 508–515, Jan. 2025.
- [35] A. J. Sathyamoorthy et al., "CoNVOI: Context-aware navigation using vision language models in outdoor and indoor environments," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2024, pp. 13837–13844.
- [36] K. Fang, F. Liu, P. Abbeel, and S. Levine, "Moka: Open-world robotic manipulation through mark-based visual prompting," in *Proc. Conf. Robot. Sci. Syst.*, 2024.
- [37] J. Achiam et al., "GPT-4 technical report," 2023, *arXiv:2303.08774*.
- [38] J. Liang et al., "GND: Global navigation dataset with multi-modal perception and multi-category traversability in outdoor campus environments," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2025.
- [39] H. Alt and M. Godau, "Computing the Fréchet distance between two polygonal curves," *Int. J. Comput. Geometry Appl.*, vol. 5, no. 01n02, pp. 75–91, 1995.
- [40] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," *IEEE Robot. Automat. Mag.*, vol. 4, no. 1, pp. 23–33, Mar. 1997.