

# DTG : Diffusion-based Trajectory Generation for Mapless Global Navigation

Jing Liang<sup>1</sup>, Amirreza Payandeh <sup>2</sup>, Daeun Song<sup>1</sup>, Xuesu Xiao<sup>2</sup> and Dinesh Manocha<sup>1</sup>

**Abstract**—We present a novel end-to-end diffusion-based trajectory generation method, DTG, for mapless global navigation in challenging outdoor scenarios with occlusions and unstructured off-road features like grass, buildings, bushes, etc. Given a distant goal, our approach computes a trajectory that satisfies the following goals: (1) minimize the travel distance to the goal; (2) maximize the traversability by choosing paths that do not lie in undesirable areas. Specifically, we present a novel Conditional RNN(CRNN) for diffusion models to efficiently generate trajectories. Furthermore, we propose an adaptive training method that ensures that the diffusion model generates more traversable trajectories. We evaluate our methods in various outdoor scenes and compare the performance with other global navigation algorithms on a Husky robot. In practice, we observe at least a 15% improvement in traveling distance and around a 7% improvement in traversability. Video and Code: <https://github.com/jingGM/DTG.git>.

## I. INTRODUCTION

Global navigation is used to compute the trajectories of robots in large-scale environments [1]–[3]. Global navigation is widely used in various tasks, such as autonomous driving [3], [4], last-mile delivery [5], [6], and search and rescue operations [7]. However, various challenges have to be addressed to successfully conduct global navigation tasks [8]–[11].

Mapless Navigation is Critical for Outdoor Global Navigation. To facilitate planning, a global map is computed for many global navigation strategies [1], [2], [12]. However, the acquisition of an accurate and detailed map poses significant challenges, particularly for outdoor navigation tasks with frequently changing environments [13] due to weather changes [14], [15], temporary construction sites [16], and hazardous areas [17]. Therefore, it is important to develop mapless navigation strategies for general outdoor scenes. Nevertheless, there are various challenges associated with mapless outdoor navigation, including traversability analysis, optimality assurance, constraints satisfaction, etc. [10], [18].

*Computing Traversable Trajectories for Robotic Navigation:* In complex outdoor scenarios, traversability analysis is critical for safe outdoor navigation because outdoor environments contain various challenging terrains, plants, buildings, trees, etc. [19]–[23]. Traditionally, perception and planning are decoupled as two different tasks [18], [24], [25], perception to detect traversable areas and planning to generate waypoints. However, the traversable maps created by these approaches may not be the most effective representation of the environment. For example, in learning-based approaches [26]–[28], the high-dimensional observations are



Fig. 1: DTG generates trajectories in traversable areas with the shortest travel distance to the target (red star). The blue and yellow boxes show the efficacy of the generated trajectories in scenarios including bushes and buildings.

usually processed into a vector to encode the traversability information. Furthermore, for mapless navigation with occlusions, estimating traversable areas behind occluding objects can be challenging. Therefore, effectively modeling the complex traversability information in outdoor scenarios is important.

*Optimal Trajectory Generation Towards Designated Goals.* Beyond ensuring traversability, navigation problems are often formulated as optimization problems [19], [29], [30], aiming to compute optimal trajectories under constraints. Some of the common optimality criteria include minimizing the running time or the traveling distance [26], [30]–[32]. However, in mapless navigation, if the goal is far away, the absence of an exact map complicates the evaluation of the remaining travel distance, e.g., whether traversing through a wider or narrower passage will result in shorter travel distance [33], [34].

*Generating Trajectories with Traversability and Optimality.* Learning-based approaches demonstrate remarkable performance in outdoor navigation tasks [10], [19], [27]. However, the selection of an effective model to accurately generate trajectories that are optimal in terms of traveling distance and to satisfy traversability constraints is challenging. While on-policy reinforcement learning-based outdoor navigation approaches [19], [35] can generate optimal trajectories, it is not safe to try failure cases, such as collision or flipping over, in complex outdoor scenarios. Many approaches apply supervised learning with different models [11], [27], [28], [36]. The diffusion model shows promising performance in different robotic applications, such as picking and pushing

<sup>1</sup>University of Maryland, College Park. <sup>2</sup>George Mason University

operations of robotic arms [37] and navigation [27]. However, the denoising procedure with U-Net [37] is still not computationally efficient for real-time navigation. However, NoMaD still requires very close subgoal images, which is not fully mapless, and the nearest subgoals need to be very close to the robot. Furthermore, the diffusion models are only trained to imitate the ground truth without any additional constraints, such as the traversability constraints.

**Main Innovations:** We introduce the diffusion mechanism in the mapless outdoor global navigation task and present a novel end-to-end approach, DTG, to generate trajectories to alleviate the challenges of complex outdoor mapless environments. Under the condition of the environmental information, the diffusion model takes a random Gaussian noise and denoises it in multiple steps to predict a traversable trajectory with short travel distance to the goal. We also demonstrate the benefits of our approach in complex outdoor scenarios with occlusions and unstructured elements. The major contributions include:

- 1) **A Novel End-to-end Diffusion-based Trajectory Generator for Global Navigation:** We apply diffusion models in the mapless outdoor global navigation task. The diffusion-based generator generates the trajectories with decent traversability and short future travel distances in the global navigation task with a distant goal (>50 meters).
- 2) **A Novel Conditional RNN (CRNN) Model for The Diffusion Model:** To make diffusion models run in real time for navigation, we propose CRNN, which takes the environment information as conditions and generates trajectories in real time for global navigation.
- 3) **Adaptive Training to Enhance Traversability:** To enhance the traversability of the generated trajectories, we propose a new method to adaptively apply traversability loss to different diffusion steps according to the historic loss of the steps.
- 4) **Performance Improvement in Global Navigation:** We demonstrate the benefits of our approach, DTG, in complex outdoor scenarios with occlusions and complex features, such as bushes, grass, and other off-road, non-traversable areas. We compare with the state-of-the-art trajectory generation approaches (ViNT [28], NoMaD [27], and MTG [11]) for outdoor global navigation and observe at least a 15% improvement in future travel distance and a 7% improvement in traversability. We also qualitatively show the benefits of our approach: around corners with occlusions DTG generates trajectories with better traversability, and around narrow spaces it can generate trajectories with shorter path lengths. Our proposed CRNN also achieves traversability and travel distances comparable to U-Net [37] with less running time and smaller a model size for real-time navigation.

## II. PRIOR WORK AND BACKGROUND

In this section, we review the related works on trajectory generation in challenging outdoor navigation.

**Outdoor Navigation:** Navigating robots in outdoor environments presents significant challenges due to occlusions, weather conditions, construction sites, and diverse and complex terrains like grass, bushes, mud, and sharp elevation changes [20]–[25]. Various strategies [19], [20], [38], [39] have been proposed to address these issues. Motion planning techniques aim for stable and safe robot movement across different terrains [19], [20], using adaptive approach [19] and reinforcement learning [38], [39] to train neural networks for waypoint generation. Global planning requires a comprehensive cost map for path planning [40], [41] and accurate robot localization [42], [43] to follow the paths. However, those map-based navigation approaches can be computationally expensive and require a significant amount of overhead to maintain the maps. To solve this issue, instead of maintaining a comprehensive map, Sridhar et al. [27], [28] and Hirose et al. [44] propose generating topological maps and using images as subgoals for navigation, but those approaches still require initial runs in the environment to gather subgoal images. Mapless navigation techniques are used to navigate without relying on maps. Giovannangeli et al. [9] and Liang et al. [19] generate actions in a mapless manner, but they focus on local planning without addressing long-distance navigation. MTG [11] represents a step forward by providing long-distance navigation trajectories, yet it doesn't optimize for the best path. In contrast, we propose a novel approach, DTG, which generates traversable trajectories with short travel distances towards a distant goal in large-scale outdoor settings without a map.

**Traversability Analysis:** Traversability analysis plays a critical role in robot navigation, distinguishing between navigable and non-navigable areas within an environment. This task is often handled by separating perception and planning [45]–[47], utilizing various sensors to assess the terrain. Cameras (RGB or RGB-D) are commonly employed to analyze the terrain [47], [48], enabling segmentation and the creation of cost maps for navigation. Similarly, Lidar sensors are used for their ability to generate elevation maps through geometric information [49], [50]. However, reliance on cost or elevation maps, while visually intuitive for humans, poses computational costs for robots due to the extra processing and encoding of the maps. To address this, end-to-end learning approaches [11], [19], [51] have been developed to encode environmental information directly into neural networks, thereby streamlining the navigation process. In our work, we also apply an end-to-end learning approach to encode the observation information.

**Trajectory Generation:** Trajectory generation in outdoor scenarios varies significantly for different applications. Some methods are designed specifically for autonomous driving [52]–[54] with LSTM [55] or Gaussian Mixture Models [52] to predict trajectories based on historical movements. Smaller robots' trajectory generation strategies [10], [28], [36] leverage Bayesian-based methods [11], [56], GANs (Generative Adversarial Networks) [57], [58] etc to compute feasible paths through complex environments. Existing methods for small robots global navigation, such as ViNT [28]

and NoMaD [27], rely on comparing current images to pre-recorded subgoal images to navigate, but they cannot recognize the perceived images with significant differences from the subgoal images or in completely unknown environments, so sophisticated choices of subgoals are necessary [28]. Moreover, they do not consider the optimality of the trajectories in terms of travel distance to the goals and require prior knowledge (subgoal images) of the environment, limiting their applicability to unknown or dynamically changing areas. MTG [11] uses CVAE [59] to generate trajectories in traversable areas, but it doesn't consider the optimality of the trajectories. Diffusion models [60], [61] have been used for robotics applications, such as picking and pushing objects [37] and navigation [27]. In our approach, we propose a novel diffusion-based mapless trajectory generator to generate optimal trajectories with short travel distances to goals and train the generation with traversabilities metrics.

### III. APPROACH

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

#### A. Problem Definition

The problem we are solving is outdoor mapless global navigation. Given a distant goal  $g \in \mathcal{O}_g$  in a large-scale environment, our model generates trajectories in the robot's traversable areas while trying to minimize the travel distance to the goal. We assume the trajectory generator ignores small and dynamic obstacles, which can be handled by the local planners.

Perceptual sensors used in our approach include a 3D LiDAR and the robot's odometer. We utilize  $C_l$  consecutive frames of LiDAR perception,  $\mathcal{O}_l$ , to capture the static and dynamic information of the environment in the robot's vicinity. The odometer provides the latest  $C_v$  consecutive frames of the robot's velocities,  $\mathcal{O}_v$ , to encode the robot's dynamic status. In this global navigation task, the goal is denoted as  $g = \{g_x, g_y\} \in \mathcal{O}_g$ . Thus, the observation contains  $\mathcal{O} = \{\mathcal{O}_l, \mathcal{O}_v, \mathcal{O}_g\}$ . Given the observation  $\mathcal{O}$ , our trajectory generator  $DTG_\theta$  generates a trajectory  $\tau = \{\mathbf{w}_1, \dots, \mathbf{w}_M\}$  that both satisfies traversability constraints and has minimum travel distance to the goal  $g$ . Here,  $\mathbf{w}_m = \{x_m, y_m\}$  represents the waypoints in the generated trajectory, which contains  $M$  waypoints in total starting from the robot. We define the travel distance of the trajectory  $\tau$  as the length of the shortest path from the last waypoint  $\mathbf{w}_M$  of  $\tau$  to the target  $g$ . For each observation  $o \in \mathcal{O}$ , we have  $DTG_\theta(o) = \tau$ . The problem can be formulated as the following equation:

$$\hat{\theta} = \arg \min_{\theta} \left( h(DTG_\theta(o), g) + \beta f(DTG_\theta(o), \tilde{\mathcal{A}}) \right) \quad (1)$$

where  $h(\cdot, \cdot)$  represents the travel-distance function between the last waypoint  $\mathbf{w}_M$  and the goal  $g$ .  $\mathcal{A}$  is the traversable area around the robot and  $\tilde{\mathcal{A}}$  represents non-traversable areas.  $\beta$  is a hyperparameter and  $f(\cdot, \cdot)$  calculates the traversability of the trajectory  $\tau$ , which is the portion of non-traversable

areas covered by the trajectory  $\tau$ . Therefore, the function achieves two targets: 1. Optimize the travel distance; and 2. Satisfy traversability constraints. The inference of our approach only takes  $\mathbf{o} \in \mathcal{O}$  as input, but to train the models, we require  $\mathcal{A}$  to calculate the traversability ground truth.

For training, DTG uses a similar traversability map as in MTG [11], where the off-road areas and buildings are not traversable, but sidewalks, pavements, and drivable roads are traversable areas. The travel-distance function  $h(\cdot, \cdot)$  can be handled by the A\* algorithm, starting from the last waypoint  $\mathbf{w}_M$  to the goal  $g$ . The travel distance is the length of the calculated A\* path. The non-traversable area  $\tilde{\mathcal{A}}$  can also be directly extracted from the traversability map.

Since we already have the traversability map and use path planning methods to calculate the travel distance of trajectories, we can directly use the trajectories as the ground truth of our model, DTG, in the training. Thus, we redefine the problem as Equation 2.

$$\hat{\theta} = \arg \min_{\theta} \left( d(DTG_\theta(o), \tau_{gt}) + \beta f(DTG_\theta(o), \tilde{\mathcal{A}}) \right), \quad (2)$$

where  $\tau_{gt}$  is the ground truth trajectory with the shortest travel distance to the goal  $g$  and lies in traversable areas.  $d(\cdot, \cdot)$  calculates the distance between the generated trajectory and the ground truth trajectory  $\tau_{gt}$ . Since the ground truth and inputs are all given, this problem can be handled by a supervised learning method.

#### B. Architecture

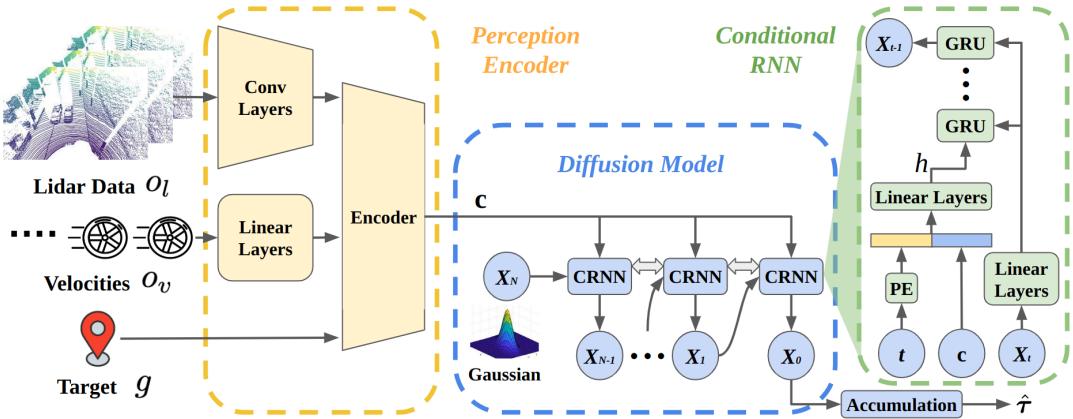
Figure 2 shows the end-to-end architecture of our approach, DTG. There are two models in the pipeline: **Perception Encoder**  $P_\theta(\cdot)$  and **Diffusion Model**  $D_\theta(\cdot)$ . Here we denote all model parameters as  $\theta$ . Details about the layer configurations are in Appendix VI [62].

1) **Perception Encoder:** The Perception Encoder encodes the LiDAR, Velocities, and Target information into a vector as the condition input of the Diffusion Model:

$$\mathbf{c} = P_\theta(o_l, o_v, g) = p_\theta^e(p_\theta^c(o_l), p_\theta^v(o_v), g), \quad (3)$$

where  $o_l \in \mathcal{O}_l$ ,  $o_v \in \mathcal{O}_v$  and  $g \in \mathcal{O}_g$ .  $p_\theta^c(\cdot)$  represents the model PointCNN [63].  $p_\theta^v(\cdot)$  is a sequence of Linear layers to process the robot's dynamic status with historic velocities. As shown in Figure 2, the Encoder layer takes the concatenated embeddings from  $p_\theta^c(o_l)$ ,  $p_\theta^v(o_v)$ , and  $g$  and encodes the embeddings to a vector,  $\mathbf{c}$ , as the condition of the Diffusion model. The Encoder,  $p_\theta^e(\cdot)$ , also composes a sequence of Linear layers.

2) **Diffusion Model:** The Diffusion model generates high-quality data by progressively denoising a Gaussian noise to some target data [60], [61], and it shows promising capabilities in robotics tasks [37]. In our approach, the diffusion model takes the conditional vector  $\mathbf{c}$  from the Perception Encoder for each observation and denoise a Gaussian distribution to a trajectory  $\hat{\tau}$ . In recent years, multiple diffusion training strategies have been proposed, including DDPM [60], DDIM [61], and VDM [64]. Because



**Fig. 2: DTG Architecture:** DTG has two models: **Perception Encoder** and **Diffusion Model**. Perception Encoder encodes the observation information,  $\mathbf{o} = \{o_l, o_v, g\}$ , to the condition vector,  $c$ . The Diffusion Model takes a Gaussian distribution under the condition  $c$ .

VDM [64] shows higher quality in data generation and better convergence than others, in this approach, we use the VDM [64] strategy to generate trajectories. In the diffusion process, neural networks (diffusion cells) are trained to learn noise or noisy data, and these two procedures are mathematically equivalent. Because of faster convergence and smaller converged loss, we use the diffusion cells to predict trajectories in our approach, DTG. Our diffusion generator contains  $N$  steps. As shown in Figure 2, from the final output ( $x_0$ ) to the Gaussian noise ( $x_N$ ), noise is introduced in each step, resulting in a noisy trajectory denoted as  $\tilde{x}_t = \sqrt{\alpha_t}x_{t-1} + \sqrt{1-\alpha_t}\epsilon$ , where  $\alpha_t$  is the noise ratio at step  $t$ , which ranges from 1 to  $N$ .  $\epsilon$  represents the noise itself. The predicted trajectory in step  $t$  is  $\hat{x}_{t-1} = R_\theta(\tilde{x}_t, t)$ , where  $R_\theta(\cdot)$  is a diffusion cell. Because the U-Net [37], [60] is very computationally intensive, we propose a novel diffusion cell to reduce the computational cost in Figure 2. Since we need the generated trajectories to remain always within traversable areas, we require the diffusion cell to also integrate environmental information. Inspired by [37], we propose a Conditional RNN (CRNN), shown on the right side of Figure 2, that takes environmental information vector  $c$  and the step number  $t$  as conditions for the generator, denoted by  $R_\theta(\tilde{x}_{t-1}, t, c)$ . Thus, we have the CRNN cell:

$$d_\theta^h(t, c) = f_\theta^2(f_\theta^1(d^p(t)), c) = \mathbf{h}, \quad (4)$$

$$r_\theta^k(\tilde{x}_t, \mathbf{h}) = d_\theta^q(\tilde{x}_t, \mathbf{h}), \quad (5)$$

$$R_\theta(\tilde{x}_t, t, c) = r_\theta^1(\tilde{x}_t, \dots, r_\theta^1(\tilde{x}_t, \mathbf{h})), \quad (6)$$

where  $d_\theta^h(\cdot)$  calculates the hidden vector  $\mathbf{h}$  for GRU cells.  $d^p(\cdot)$  represents Sinusoidal positional embedding.  $f_\theta^1(\cdot)$  and  $f_\theta^2(\cdot)$  are all Linear layers.  $k \in \{1, \dots, K\}$  represents the steps in the CRNN model. Each step is  $r_\theta^k(\cdot)$ .  $d_\theta^q(\cdot)$  is the GRU cell and  $\mathbf{h}$  is the hidden vector for the GRU cell. We also compare with U-Net [37] as the conditional encoder and show the results in Table I demonstrating that we have comparable results, in terms of traversability and travel distance, but significantly less computational cost.

Then we have the sampled trajectory:

$$\hat{x}_t = R_\theta(\dots R_\theta(\xi, N, c), 1, c), \quad (7)$$

where  $\xi$  is sampled from a random Gaussian distribution,  $\mathcal{N}(\mu, \nu)$ . As shown in Figure 2, all CRNN models share the same parameters. The output of the diffusion model  $\hat{x}_0$  composes a sequence of  $\{\Delta x_m, \Delta y_m\}$ , and the waypoint positions  $w_m = \{x_m, y_m\} \in \hat{\tau}$  are calculated by accumulating the incremental distances.

### C. Training Strategy

According to the problem defined in Section III-A, we have two targets to achieve: 1. Reduce the distance between the generated trajectories and the ground truths, and 2. Minimize the portion of generated trajectories in non-traversable areas. Since our approach is an end-to-end model, we jointly train both targets with an adaptive training strategy:

1) *Train the generated trajectories to align with the ground truth paths, which have the shortest travel distance:* As mentioned in Section III-B.2, our approach uses the training of predicting trajectories. According to the diffusion loss in [64], we formulate our diffusion loss as

$$\mathcal{L}_d = \mathbb{E}_{t, \epsilon_d} ((\text{SNR}(t-1) - \text{SNR}(t)) \|\hat{\tau}_t - \tau_{gt}\|_2^2), \quad (8)$$

where  $t \in \{1, \dots, N\}$  and  $\epsilon_d \in \mathcal{N}(0, \mathbf{I})$ .  $\tau_{gt}$  is the ground truth trajectory because optimizing VDM [64] boils down to predicting the original ground truth.  $\text{SNR}(t) = \frac{\bar{\alpha}_t}{1-\bar{\alpha}_t}$ .  $\bar{\alpha}_t = \prod_{i=1}^t \alpha_i$  is defined the same as in [60]. To simplify the loss function, because  $\text{SNR}(t)$  is monotonically decreasing,  $(\text{SNR}(t-1) - \text{SNR}(t))$  is always positive. We find similar training results after removing this term during training. Thus, the loss function is simplified as:

$$\mathcal{L}_d = \mathbb{E}_{t, \epsilon_d} \|\hat{\tau}_t - \tau_{gt}\|_2^2. \quad (9)$$

2) *Adaptive Training of Diffusion Models:* The current loss function only trains the diffusion model to imitate the ground truth trajectories, and we still need to add constraints to train the generated trajectories only in traversable areas. However, because the diffusion model denoises the Gaussian distribution step by step, we cannot directly add the traversability constraint in each step or the training gets sabotaged. Thus, we propose an adaptive strategy to add the traversability constraints.

$$\mathcal{L}_t = \mathbb{E}_{t \in \mathbf{b}_t} \exp \left( 1 - \frac{1}{M} \sum_{m=1}^M \min d(\tilde{\mathcal{A}}, w_m) \right), \quad (10)$$

where  $d(\cdot)$  calculates the distance between the current waypoint  $w_m$  in  $\hat{\tau}_t$  and its nearest non-traversable area. The distance function clips the value to  $[0-1]$  meters. We sample  $t$  from the diffusion step buffer  $\mathbf{b}_t$ , which stores the currently available diffusion steps. As shown in Algorithm 1,  $m_\theta(\cdot)$  is the neural network model and  $\theta$  represents the parameters.  $i$  is the index of training epochs and  $N_t = 10$  is the maximum step number to apply to the traversability loss.  $l_d(t-1)$  calculate the average of 5 last recorded diffusion loss values at step  $t-1$ . If the average loss value is smaller than the threshold  $H_d$ , we add this step to the step buffer  $\mathbf{b}_t$ . We increase the range of diffusion steps incrementally to adapt the training of traversability to the training of ground truth trajectories. Finally, we have the total loss function for DTG:  $\mathcal{L} = \mathcal{L}_d + \beta \mathcal{L}_t$ , where  $\beta$  is a hyperparameter.

---

**Algorithm 1** Adaptive training schedule of DTG: adaptively apply the traversability loss to DTG.

---

```

Require:  $N_t \leftarrow 10$ 
Require:  $\mathbf{b}_t = \{ \}$ 
for  $i \leftarrow 0$  to Total Epochs do
     $t = \text{RandomSample}(0, N-1)$ 
     $\hat{\mathbf{x}}_t = m_\theta(\mathbf{o}, t)$ 
    if  $i \geq 1$  and  $t < N_t$  and  $l_d(t-1) < H_d$  then
        Add  $t-1$  to  $\mathbf{b}_t$ 
    end if
    if  $|\mathbf{b}_t| > 0$  and  $t \in \mathbf{b}_t$  then
         $\mathcal{L} = \mathcal{L}_d(\hat{\mathbf{x}}_t, \tau_{gt}, t) + \beta \mathcal{L}_t(\hat{\mathbf{x}}_t, \tilde{\mathcal{A}})$ 
    else
         $\mathcal{L} = \mathcal{L}_d(\hat{\mathbf{x}}_t, \tau_{gt})$ 
    end if
end for

```

---

#### IV. EXPERIMENTS

In this section, we discuss the details of the implementation, comparisons, and ablation studies of this approach. The experiments are designed to demonstrate the benefits of our innovations:

- 1) **Evaluate the novel end-to-end model, DTG, in maples outdoor global navigation:** We compare our approach, DTG, with SOTA outdoor navigation algorithms, including MTG [11], NoMaD [27], and ViNT [28] in both a testing dataset and challenging outdoor real-world scenarios. The real-world experiment is achieved by combining the lower-level motion planner DWA [65]. The details of the real-world experiment and the results are in Appendix VI-A [62].
- 2) **Evaluate the efficacies of Diffusion model, CRNN diffusion cells, and adaptive traversability loss for training:** For each innovation, we conduct ablation studies to demonstrate the benefit and effect of the components. The diffusion mechanism is visualized in Appendix VI-B [62]

#### A. Implementation

As described in Section III, our major perceptual sensor is a 3Hz 3D Velodyne Lidar (VLP-16) with 16 channels. The inputs of this approach contain  $C_l = 3$  consecutive frames of Lidar and  $C_v = 20$  consecutive frames of historic velocities. The goal is set by converting the GPS value to meters. During training, the goals are randomly selected within 60 meters; for testing, the goals are selected beyond 50 meters. The trajectory generator, DTG, generates  $M = 16$  waypoints in each trajectory. The voxelization radius of the PointCNN in the Perception Encoder is 0.08m in this approach. The training data is the same as MTG [11], which is collected by a Husky robot. The training and evaluation datasets are in different areas as shown in [11]. Velocities are collected from a 10Hz odometer. To keep all the perceptive information in the same time period, we select consecutive 10 frames of velocities as input. During training, the ground truth trajectories, generated by the A\* algorithm with the shortest travel distance to the goal, are thresholded 15 meters from the robot's position, and have 16 waypoints each. The training and evaluation are processed in a computer with an NVIDIA RTX A5000 GPU and an Intel Xeon(R) W-2255 CPU, and the real-world experiment is executed on a laptop with an Intel i7 CPU and one Nvidia GTX 1080 GPU. In the real-world experiment, the diffusion model generates trajectories in 5Hz. The network details of the architecture are in Appendix VI-C [62].

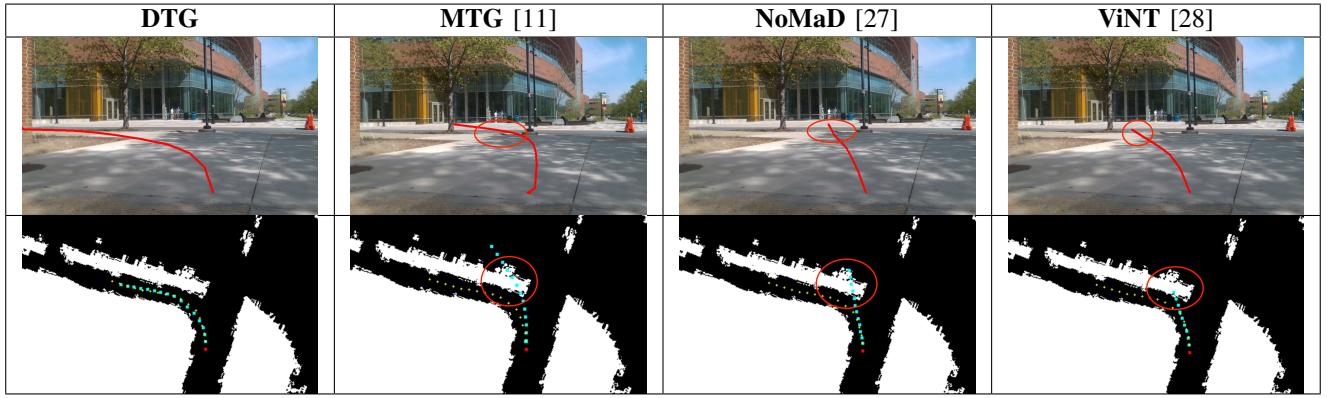
#### B. Evaluation

In this section, we qualitatively and quantitatively evaluate DTG compared with different state-of-the-art methods and modified versions for ablation study. In the experiment, we compare DTG with ViNT [28], NoMaD [27], and MTG [11]. Because NoMaD and ViNT require a sequence of images from the start position to the goal, we run the robot in the environment first and collect the images. Each consecutive pair of images has a distance of around 1 meter. MTG generates multiple trajectories, and we use the same method as the experiment in MTG [11] to choose the best trajectory. The evaluation metrics include: Traversability, Distance Ratio, Inference Time, and Model Size.

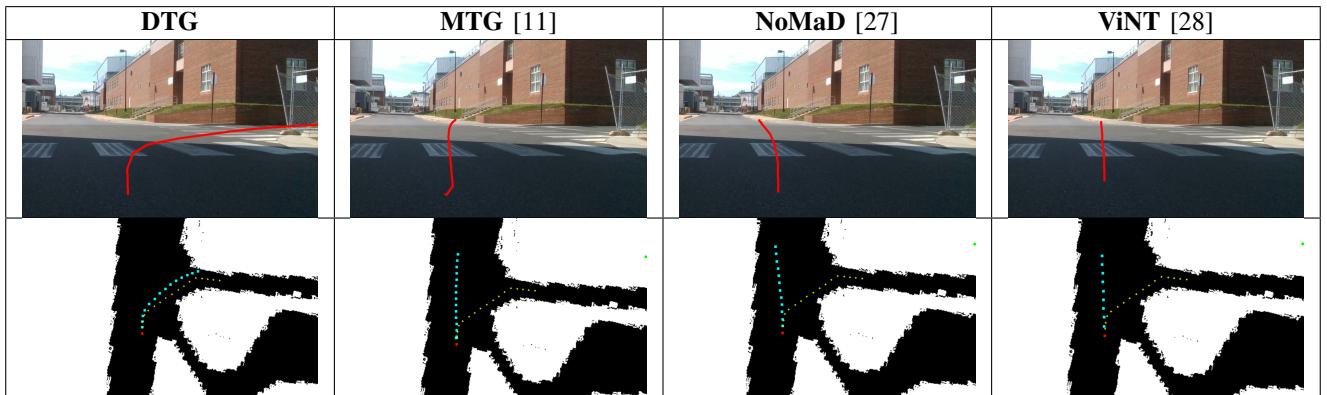
**Traversability:** Given a trajectory  $\hat{\tau}$ , the traversability is calculated as Equation 11. The  $c(\cdot)$  tells if the waypoint  $w_m$  is in the traversable area  $\mathcal{A}$ . For K scenarios, we have the average traversability of the approach:  $\frac{1}{K} \sum_{i=1}^K tr(\mathcal{A}, \hat{\tau})$ .

$$tr(\mathcal{A}, \hat{\tau}) = \bigcap_{m=1}^M c(\mathcal{A}, w_m), \quad w_m \in \hat{\tau}. \quad (11)$$

**Distance Ratio:** The distance ratio is to evaluate the future travel distance of the trajectory to the goal and it measures the ratio of the travel distance of the trajectory w.r.t. the shortest travel distance from the robot's position to the goal. Therefore, given the trajectory length  $|\hat{\tau}|$ , the travel distance  $h_c$  from the robot's position, and the travel distance  $h_t$  from the last waypoint  $w_M \in \hat{\tau}$ , the distance ratio is defined as Equation 12. The ratio is trajectory



**Fig. 3: Traversability Analysis in Challenging Occluded Environment:** The top row shows the generated trajectories (red) in the camera view. The bottom row shows the top-down view of the traversability map. The cyan color represents the generated trajectories, and the yellow color represents the most heuristic trajectory to the goal. DTG can generate trajectories w.r.t. the geometric shape of the traversable areas, but other approaches cannot generate fully traversable trajectories; the non-traversable parts are marked by red circles.



**Fig. 4: Travel-Distance Analysis in Challenging Narrow Passage Environment:** The goal is behind the building. DTG generates a trajectory in a narrower space instead of the wide main road, leading to a shorter travel distance to the goal.

length independent, so we can use it to compare trajectory generators with different trajectory lengths.

$$hr(\hat{\tau}) = 1 - \frac{|h_t - h_c|}{2|\hat{\tau}|} \quad (12)$$

**Comparisons:** This section shows the efficacy of DTG for mapless outdoor global navigation. From Table I, we observe our approach, DTG, outperforms other SOTA approaches. Specifically, we achieve by at least 15% in distance ratio and 12.6% in traversability improvement compared with ViNT and NoMaD. NoMaD and ViNT only take RGB images, so the robot is not very robust in generating trajectories only in traversable areas. NoMaD is better than ViNT w.r.t. inference time but slightly worse in traversability. Our approach outperforms MTG around 7% in traversability and 16% in distance ratio. The MTG has relatively good traversability but cannot choose the trajectory with the best distance ratio because in MTG the trajectories are generated by only comparing the straight distance to the goal instead of estimating the real travel distance of the trajectory. DTG also outperforms NoMaD and ViNT in inference time by 0.11s and 0.56s, but the model size is bigger than ViNT, NoMaD, and MTG.

As shown in Figures 3 and 4, we provide the qualitative explanation of how our approach outperforms other SOTA

methods. The top row shows the generated trajectory in the camera and the bottom row shows the trajectory in the traversability map, where cyan represents the generated trajectory and yellow dots are the ground truth trajectories with the shortest travel distance to the goal. From the two figures, we observe NoMaD and ViNT generate shorter trajectories than MTG or DTG. Figure 3 shows a challenging occluded environment around a corner; DTG can generate the trajectory aligned with the geometric shape of the traversable areas, but MTG and NoMaD do not perform well in challenging corner situations. In Figure 4, the goal is behind the building and the ground truth trajectory lies in a narrow passage. The scenario is challenging in terms of travel distance estimation. DTG can still generate a trajectory to the target in the narrow passage, while other approaches all choose trajectories in the wider main road.

**Ablation Study:** To evaluate the capability of different components of our innovations, we compare DTG with the modified versions without traversability loss during training, changing our Conditional RNN to the regular U-Net model and changing the generative model from the diffusion model to CVAE [59]. From Table I, our DTG has the best heuristic compared with other ablation studies. The U-Net is much larger than our proposed CRNN with more than 989.51Mb

Evaluation	Input Modality	Distance Ratio (%)	Traversability (%)	Inference Time (s)	Model Size (Mb)
ViNT	RGB	66.02	79.02	0.69	113.49
NoMaD	RGB	64.54	77.86	0.24	72.67
MTG	Lidar	80.78	83.11	0.01	101.61
CVAE	Lidar	92.26	85.89	0.01	113.96
DTG/t	Lidar	93.12	85.57	0.13	128.38
DTGU-Net	Lidar	92.94	90.74	2.09	1117.89
DTGCRNN	Lidar	93.61	89.00	0.13	128.38

TABLE I: **Quantitative Results:** Our approach achieves at least a 15% improvement in distance ratio and around 7% increase in traversability over other approaches. Our innovative components, CRNN, adaptive traversability training, and end-to-end diffusion-based generator also show effective improvement in global navigation task.

and is also much slower than all other approaches, but because the U-Net model is larger than CRNN, it can encode information better and generates trajectories with slightly better traversability. Our novel model CRNN is faster and smaller than U-Net, but we achieve comparable distance ratios and traversability in trajectory generation. The DTG/t shows the model without training traversability loss. Obviously, it has the worst traversability. The CVAE model is smaller, but the generative capability is not as good as diffusion models. Its distance ratio and traversability of the generated trajectories are worse than DTG.

## V. CONCLUSION, LIMITATIONS, AND FUTURE WORK

We present a novel end-to-end diffusion-based trajectory generator for mapless global navigation and demonstrate the innovations of the end-to-end approach and the efficacy of the different innovative components in both evaluation dataset and the real-world experiments. We achieve at least a 15% improvement in distance ratio and a 7% improvement in traversability over other SOTA approaches.

There are also limitations of the approach, DTG. Because the trajectory generator generates trajectories in real-time, there should be some mechanism to smartly choose the best trajectory during navigation, e.g., estimating the feasibility and confidence of the trajectory.

## ACKNOWLEDGMENT

This work was supported in part by ARO Grants W911NF2310046, W911NF2310352 and U.S. Army Cooperative Agreement W911NF2120076

## REFERENCES

- [1] 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 Journal for Science and Engineering*, vol. 47, no. 3, pp. 3633–3647, 2022.
- [2] P. Gao, Z. Liu, Z. Wu, and D. Wang, “A global path planning algorithm for robots using reinforcement learning,” in *2019 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. IEEE, 2019, pp. 1693–1698.
- [3] T. Ort, L. Paull, and D. Rus, “Autonomous vehicle navigation in rural environments without detailed prior maps,” in *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2018, pp. 2040–2047.
- [4] E. Yurtsever, J. Lambert, A. Carballo, and K. Takeda, “A survey of autonomous driving: Common practices and emerging technologies,” *IEEE access*, vol. 8, pp. 58 443–58 469, 2020.
- [5] T. Hoffmann and G. Prause, “On the regulatory framework for last-mile delivery robots,” *Machines*, vol. 6, no. 3, p. 33, 2018.
- [6] C. Chen, E. Demir, Y. Huang, and R. Qiu, “The adoption of self-driving delivery robots in last mile logistics,” *Transportation research part E: logistics and transportation review*, vol. 146, p. 102214, 2021.
- [7] A. Davids, “Urban search and rescue robots: from tragedy to technology,” *IEEE Intelligent systems*, vol. 17, no. 2, pp. 81–83, 2002.
- [8] G. Huang, A. Rad, and Y. Wong, “Online slam in dynamic environments,” in *ICAR’05. Proceedings., 12th International Conference on Advanced Robotics, 2005*. IEEE, 2005, pp. 262–267.
- [9] C. Giovannangeli, P. Gaussier, and G. Désilles, “Robust mapless outdoor vision-based navigation,” in *2006 IEEE/RSJ international conference on intelligent robots and systems*. IEEE, 2006, pp. 3293–3300.
- [10] D. Shah and S. Levine, “Viking: Vision-based kilometer-scale navigation with geographic hints,” *arXiv preprint arXiv:2202.11271*, 2022.
- [11] J. Liang, P. Gao, X. Xiao, A. J. Sathyamoorthy, M. Elnoor, M. Lin, and D. Manocha, “Mtg: Mapless trajectory generator with traversability coverage for outdoor navigation,” *arXiv preprint arXiv:2309.08214*, 2023.
- [12] M. Psotka, F. Duchon, M. Roman, T. Michal, and D. Michal, “Global path planning method based on a modification of the wavefront algorithm for ground mobile robots,” *Robotics*, vol. 12, no. 1, p. 25, 2023.
- [13] L. Wijayathunga, A. Rassau, and D. Chai, “Challenges and solutions for autonomous ground robot scene understanding and navigation in unstructured outdoor environments: A review,” *Applied Sciences*, vol. 13, no. 17, p. 9877, 2023.
- [14] Y. Zhang, R. Ge, L. Lyu, J. Zhang, C. Lyu, and X. Yang, “A virtual end-to-end learning system for robot navigation based on temporal dependencies,” *IEEE Access*, vol. 8, pp. 134 111–134 123, 2020.
- [15] T. Ort, I. Gilitschenski, and D. Rus, “Autonomous navigation in inclement weather based on a localizing ground penetrating radar,” *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3267–3274, 2020.
- [16] I. Jeong, Y. Jang, J. Park, and Y. K. Cho, “Motion planning of mobile robots for autonomous navigation on uneven ground surfaces,” *Journal of Computing in Civil Engineering*, vol. 35, no. 3, p. 04021001, 2021.
- [17] W. Eom, J. Park, and J. Lee, “Hazardous area navigation with temporary beacons,” *International Journal of Control, Automation and Systems*, vol. 8, no. 5, pp. 1082–1090, 2010.
- [18] S. M. LaValle, *Planning algorithms*. Cambridge university press, 2006.
- [19] J. Liang, K. Weerakoon, T. Guan, N. Karapetyan, and D. Manocha, “Adaptiveon: Adaptive outdoor local navigation method for stable and reliable actions,” *IEEE Robotics and Automation Letters*, vol. 8, no. 2, pp. 648–655, 2022.
- [20] 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 Robotics and Automation Letters*, 2023.
- [21] X. Xiao, J. Biswas, and P. Stone, “Learning inverse kinodynamics for accurate high-speed off-road navigation on unstructured terrain,” *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 6054–6060, 2021.
- [22] H. Karnan, K. S. Sikand, P. Atreya, S. Rabiee, X. Xiao, G. Warnell, P. Stone, and J. Biswas, “Vi-ikd: High-speed accurate off-road navigation using learned visual-inertial inverse kinematics,” in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 3294–3301.
- [23] A. Pokhrel, A. Datar, M. Nazeri, and X. Xiao, “Cahsor: Competence-aware high-speed off-road ground navigation in se (3),” *arXiv preprint arXiv:2402.07065*, 2024.
- [24] J. Canny, *The complexity of robot motion planning*. MIT press, 1988.
- [25] D. Manocha, *Algebraic and numeric techniques in modeling and robotics*. University of California, Berkeley, 1992.
- [26] J. Liang, U. Patel, A. J. Sathyamoorthy, and D. Manocha, “Crowdsteer: Realtime smooth and collision-free robot navigation in densely

- crowded scenarios trained using high-fidelity simulation,” in *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, 2021, pp. 4221–4228.
- [27] A. Sridhar, D. Shah, C. Glossop, and S. Levine, “Nomad: Goal masked diffusion policies for navigation and exploration,” *arXiv preprint arXiv:2310.07896*, 2023.
- [28] D. Shah, A. Sridhar, N. Dashora, K. Stachowicz, K. Black, N. Hirose, and S. Levine, “Vint: A foundation model for visual navigation,” *arXiv preprint arXiv:2306.14846*, 2023.
- [29] J. Liang, Y.-L. Qiao, T. Guan, and D. Manocha, “Of-vo: Efficient navigation among pedestrians using commodity sensors,” *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 6148–6155, 2021.
- [30] D. Malyuta, T. P. Reynolds, M. Szmuk, T. Lew, R. Bonalli, M. Pavone, and B. Acikmese, “Convex optimization for trajectory generation: A tutorial on generating dynamically feasible trajectories reliably and efficiently,” *IEEE Control Systems Magazine*, vol. 42, no. 5, pp. 40–113, 2022.
- [31] A. Gasparetto, P. Boscaroli, A. Lanzutti, and R. Vidoni, “Path planning and trajectory planning algorithms: A general overview,” *Motion and Operation Planning of Robotic Systems: Background and Practical Approaches*, pp. 3–27, 2015.
- [32] J. R. Sanchez-Ibanez, C. J. Perez-del Pulgar, and A. Garcia-Cerezo, “Path planning for autonomous mobile robots: A review,” *Sensors*, vol. 21, no. 23, p. 7898, 2021.
- [33] Q. Wu, J. Wang, J. Liang, X. Gong, and D. Manocha, “Image-goal navigation in complex environments via modular learning,” *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 6902–6909, 2022.
- [34] Y. Jang, J. Baek, and S. Han, “Hindsight intermediate targets for map-less navigation with deep reinforcement learning,” *IEEE Transactions on Industrial Electronics*, vol. 69, no. 11, pp. 11816–11825, 2021.
- [35] O. Doukhi and D. J. Lee, “Deep reinforcement learning for autonomous map-less navigation of a flying robot,” *IEEE Access*, vol. 10, pp. 82 964–82 976, 2022.
- [36] G. Kahn, P. Abbeel, and S. Levine, “Land: Learning to navigate from disengagements,” *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 1872–1879, 2021.
- [37] C. Chi, S. Feng, Y. Du, Z. Xu, E. Cousineau, B. Burchfiel, and S. Song, “Diffusion policy: Visuomotor policy learning via action diffusion,” *arXiv preprint arXiv:2303.04137*, 2023.
- [38] Z. Xu, G. Dhamankar, A. Nair, X. Xiao, G. Warnell, B. Liu, Z. Wang, and P. Stone, “Applr: Adaptive planner parameter learning from reinforcement,” in *2021 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2021, pp. 6086–6092.
- [39] Z. Xu, X. Xiao, G. Warnell, A. Nair, and P. Stone, “Machine learning methods for local motion planning: A study of end-to-end vs. parameter learning,” in *2021 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*. IEEE, 2021, pp. 217–222.
- [40] S. J. Fusic, G. Kanagaraj, K. Hariharan, and S. Karthikeyan, “Optimal path planning of autonomous navigation in outdoor environment via heuristic technique,” *Transportation research interdisciplinary perspectives*, vol. 12, p. 100473, 2021.
- [41] J. Li, H. Qin, J. Wang, and J. Li, “Openstreetmap-based autonomous navigation for the four wheel-legged robot via 3d-lidar and ccd camera,” *IEEE Transactions on Industrial Electronics*, vol. 69, no. 3, pp. 2708–2717, 2021.
- [42] N. Akai, L. Y. Morales, E. Takeuchi, Y. Yoshihara, and Y. Ninomiya, “Robust localization using 3d ndt scan matching with experimentally determined uncertainty and road marker matching,” in *2017 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2017, pp. 1356–1363.
- [43] S. Lowry, N. Sürderhauf, P. Newman, J. J. Leonard, D. Cox, P. Corke, and M. J. Milford, “Visual place recognition: A survey,” *ieee transactions on robotics*, vol. 32, no. 1, pp. 1–19, 2015.
- [44] N. Hirose, D. Shah, A. Sridhar, and S. Levine, “Exaug: Robot-conditioned navigation policies via geometric experience augmentation,” in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 4077–4084.
- [45] D. Fan, K. Otsu, Y. Kubo, A. Dixit, J. Burdick, and A.-A. Aghamohammadi, “Step: Stochastic traversability evaluation and planning for risk-aware off-road navigation,” *arXiv preprint arXiv:2103.02828*, 2021.
- [46] M. V. Gasparino, A. N. Sivakumar, Y. Liu, A. E. Velasquez, V. A. Higuti, J. Rogers, H. Tran, and G. Chowdhary, “Wayfast: Navigation with predictive traversability in the field,” *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 10 651–10 658, 2022.
- [47] F. Jonas, M. Matias, C. Nived, C. Cesar, F. Maurice, and H. Marco, “Fast traversability estimation for wild visual navigation,” in *Proceedings of Robotics: Science and Systems*, 2023. [Online]. Available: <https://arxiv.org/pdf/2305.08510.pdf>
- [48] S. Hosseinpoor, J. Torresen, M. Mantelli, D. Pitti, M. Kolberg, R. Maffei, and E. Prestes, “Traversability analysis by semantic terrain segmentation for mobile robots,” in *2021 IEEE 17th International Conference on Automation Science and Engineering (CASE)*. IEEE, 2021, pp. 1407–1413.
- [49] H. Xue, H. Fu, L. Xiao, Y. Fan, D. Zhao, and B. Dai, “Traversability analysis for autonomous driving in complex environment: A lidar-based terrain modeling approach,” *Journal of Field Robotics*, vol. 40, no. 7, pp. 1779–1803, 2023.
- [50] P. Fankhauser, M. Bloesch, and M. Hutter, “Probabilistic terrain mapping for mobile robots with uncertain localization,” *IEEE Robotics and Automation Letters (RA-L)*, vol. 3, no. 4, pp. 3019–3026, 2018.
- [51] X. Xiao, B. Liu, G. Warnell, and P. Stone, “Motion planning and control for mobile robot navigation using machine learning: a survey,” *Autonomous Robots*, vol. 46, no. 5, pp. 569–597, 2022.
- [52] B. Varadarajan, A. Hefny, A. Srivastava, K. S. Refaat, N. Nayakanti, A. Cornman, K. Chen, B. Douillard, C. P. Lam, D. Anguelov *et al.*, “Multipath++: Efficient information fusion and trajectory aggregation for behavior prediction,” in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 7814–7821.
- [53] N. Nayakanti, R. Al-Rfou, A. Zhou, K. Goel, K. S. Refaat, and B. Sapp, “Wayformer: Motion forecasting via simple & efficient attention networks,” in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 2980–2987.
- [54] S. Shi, L. Jiang, D. Dai, and B. Schiele, “Motion transformer with global intention localization and local movement refinement,” *Advances in Neural Information Processing Systems*, vol. 35, pp. 6531–6543, 2022.
- [55] F. Alché and A. de La Fortelle, “An lstm network for highway trajectory prediction,” in *2017 IEEE 20th international conference on intelligent transportation systems (ITSC)*. IEEE, 2017, pp. 353–359.
- [56] M. Yang, M. Pei, and Y. Jia, “Online maximum a posteriori tracking of multiple objects using sequential trajectory prior,” *Image and Vision Computing*, vol. 94, p. 103867, 2020.
- [57] A. Gupta, J. Johnson, L. Fei-Fei, S. Savarese, and A. Alahi, “Social gan: Socially acceptable trajectories with generative adversarial networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 2255–2264.
- [58] A. Payandeh, K. T. Baghaei, P. Fayyazsanavi, S. B. Ramezani, Z. Chen, and S. Rahimi, “Deep representation learning: Fundamentals, technologies, applications, and open challenges,” *IEEE Access*, vol. 11, pp. 137 621–137 659, 2023.
- [59] K. Sohn, H. Lee, and X. Yan, “Learning structured output representation using deep conditional generative models,” *Advances in neural information processing systems*, vol. 28, 2015.
- [60] J. Ho, A. Jain, and P. Abbeel, “Denoising diffusion probabilistic models,” *Advances in neural information processing systems*, vol. 33, pp. 6840–6851, 2020.
- [61] J. Song, C. Meng, and S. Ermon, “Denoising diffusion implicit models,” *arXiv preprint arXiv:2010.02502*, 2020.
- [62] L. Jing, P. Amirreza, S. Daeun, X. Xuesu, and M. Dinesh, “Supplement,” 2024. [Online]. Available: <https://github.com/jingGM/DTG.git>
- [63] B.-S. Hua, M.-K. Tran, and S.-K. Yeung, “Pointwise convolutional neural networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 984–993.
- [64] D. Kingma, T. Salimans, B. Poole, and J. Ho, “Variational diffusion models,” *Advances in neural information processing systems*, vol. 34, pp. 21 696–21 707, 2021.
- [65] D. Fox, W. Burgard, and S. Thrun, “The dynamic window approach to collision avoidance,” *IEEE Robotics & Automation Magazine*, vol. 4, no. 1, pp. 23–33, 1997.

Real-world Experiment	Input Modality	Traveling Distance	Human Interferes (/10)	Reached Goal
ViNT	RGB	66.61	0.2	No
NoMaD	RGB	80.18	0.2	No
MTG	Lidar	240.73	0.7	Yes
DTG	Lidar	212.91	0.2	Yes

TABLE II: DTG and MTG achieves the goal but NoMaD and ViNT cannot. Our approach, DTG, has less traveling distance and also fewer human interferes.

## VI. APPENDIX

### A. Real-world Experiments

The real-world experiment is implemented with a Husky robot, which has the biggest velocity of 1m/s. The real-world experiment is applied in the scenarios with curbs, grass, buildings, etc. Global navigation requires long-range navigation, and in our experiment, the traveling distance between the start and the goal positions is around 200 meters, and the environment is as shown in Figure 5. The robot uses GPS to localize and detect if it arrives at the goal, which is within a radius of 20 meters from the goal GPS, considering the accuracy of the GPS device is around 20 meters around buildings. The pipeline contains two parts: a trajectory generator and a motion planner. Trajectory generators take observation data and output a trajectory and the motion planner, DWA [65], handles low-level collision avoidance and follows the waypoints from the generated trajectories. In the experiment, we fine-tune the trajectory publishing frequency of each algorithm to make them perform the best. We realize that our method doesn't require a high frequency of publishing trajectories because our trajectory is longer and has better quality w.r.t. the traversability, as shown in Video [62] as qualitative results. We also quantitatively analyze the results in Table II. We calculate the travel distance until the robot achieves the goal or loses the traction of the topological map for NoMad and ViNT. The Human Interferes counts the times of human interaction with the robot when the robot is in collision or runs into non-traversable areas. We observe that DTG and MTG can reach the goal, but NoMaD and ViNT easily lose track of the topological map in the turning scenarios. Compared with MTG, our approach, DTG, has fewer human interferes and a shorter distance to the goal.

### B. Diffusion Mechanism

To visualize the mechanism of the diffusion model in DTG, we show the output of the diffusion models  $\{\Delta x_m, \Delta y_m\}$  in Figure 6. In Step 0, the increment distances are random, but after several steps, they become more concentrated and form proper shapes.

### C. Details of the Architecture

The details of the Perception Encoder are shown as Figure 7.

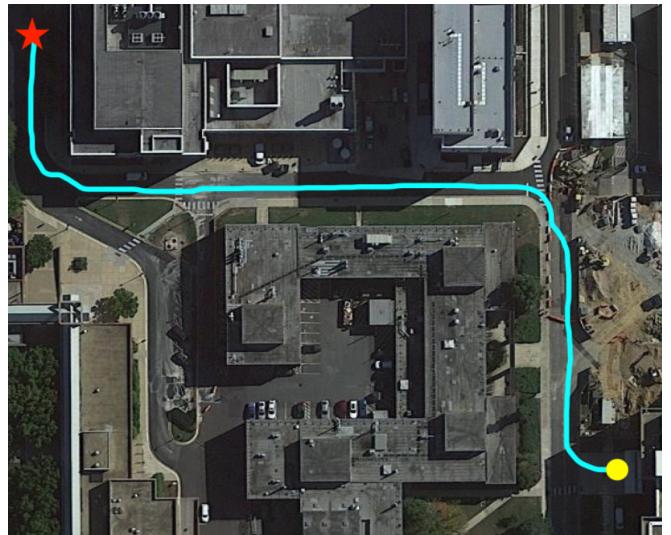


Fig. 5: The red star indicates the start position and the yellow circle indicates the goal. The example trajectory from the start to the goal is the blue path, with around a 200-meter travel distance.

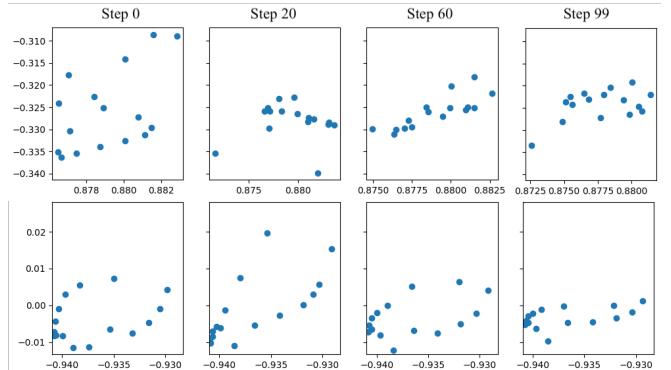


Fig. 6: Diffusion Steps: In each generated trajectory, there are 16 waypoints. This figure shows the output values of the diffusion model; those values are distance increments for each waypoint (not waypoint positions). In the beginning steps, the output contains lots of noise; in later steps, the diffusion model denoises the increments, and those values are in a more concentrated area and reasonable shape.

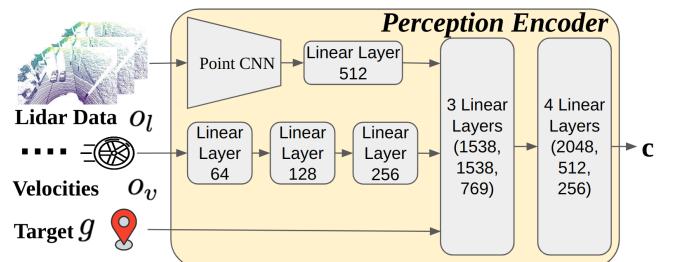


Fig. 7: The details of the Perception Encoder.



Fig. 8: Failure Case: the model confuses when the both sides of curbs have similar flatness.

#### D. Failure Cases

Considering the Lidar point clouds are sparse and because of the data quality that the curbs are not well segmented, the model is not good at detecting the curbs where both sides of the curbs have very similar flatness and the generated trajectories may lie on the curbs as Figure 8.