

FM-Planner: Foundation Model Guided Path Planning for Autonomous Drone Navigation

Jiapeng Xiao, Cheng Wen Tsao, Yuhang Zhang and Mir Feroskhan, *Member, IEEE*

arXiv:2505.20783v1 [cs.RO] 27 May 2025

Abstract—Path planning is a critical component in autonomous drone operations, enabling safe and efficient navigation through complex environments. Recent advances in foundation models, particularly large language models (LLMs) and vision-language models (VLMs), have opened new opportunities for enhanced perception and intelligent decision-making in robotics. However, their practical applicability and effectiveness in global path planning remain relatively unexplored. This paper proposes foundation model-guided path planners (FM-Planner) and presents a comprehensive benchmarking study and practical validation for drone path planning. Specifically, we first systematically evaluate eight representative LLM and VLM approaches using standardized simulation scenarios. To enable effective real-time navigation, we then design an integrated LLM-Vision planner that combines semantic reasoning with visual perception. Furthermore, we deploy and validate the proposed path planner through real-world experiments under multiple configurations. Our findings provide valuable insights into the strengths, limitations, and feasibility of deploying foundation models in real-world drone applications and providing practical implementations in autonomous flight. Project site: <https://github.com/NTU-ICG-FM-Planner>.

Index Terms—foundation model, embodied artificial intelligence, path planning, autonomous navigation

I. INTRODUCTION

AUTONOMOUS navigation capabilities are essential for drones across diverse applications, including search and rescue operations [1], parcel delivery [2], infrastructure inspection [3], and environmental monitoring [4]. Effective path planning is a critical element of autonomous flight, ensuring safe and efficient navigation by generating trajectories that guide drones through dynamic, cluttered, and often unpredictable environments. Traditional approaches to drone path planning, such as the A* algorithm [5] and Rapidly-exploring Random Trees (RRT) [6], have been widely adopted due to their simplicity but face challenges in dynamic or high-dimensional environments where adaptability and semantic awareness are required. Optimization-based methods, such as Fast-Planner [7] and EGO-Planner [8] address some of these limitations by explicitly formulating path planning as constrained optimization problems with desired objectives. Nevertheless, these methods typically rely on accurate models of the environment and robot dynamics, making them less flexible in unstructured or rapidly changing environments.

J. Xiao, C. W. Tsao, Y. Zhang, and M. Feroskhan are with the School of Mechanical and Aerospace Engineering, Nanyang Technological University, Singapore 639798, Singapore (e-mail: jiapeng001@e.ntu.edu.sg; tsao0002@e.ntu.edu.sg; yuhang004@e.ntu.edu.sg; mir.feroskhan@ntu.edu.sg). (*Corresponding author: Mir Feroskhan.*)

Recently, learning-based approaches have shown significant promise for their ability to handle complex path planning tasks without explicit modeling of dynamics and optimization [9]. Imitation learning (IL) and deep reinforcement learning (DRL) have been applied to achieve agile navigation [10], [11], perception-aware flight [12], championship-level drone racing [13] and collaborative navigation [14]–[16] and formation maneuvering [17], [18]. While these methods demonstrate strong performance within trained environments, they require large amounts of domain-specific data for each task and often struggle to generalize to unseen scenarios.

The emergence of foundation models, notably large language models (LLMs) [19] and vision-language models (VLMs) [20], has significantly advanced the field of artificial intelligence by providing powerful tools for enhanced perception, reasoning, and decision-making across diverse applications, such as design [21], [22], monitoring [23] and disassembly [24]. Foundation models are pretrained on large-scale and varied datasets, enabling them to generalize effectively across different tasks with minimal fine-tuning. Recent robotics applications leveraging foundation models include robotic manipulation [25], navigation [26], and human-robot interaction [27]. Despite these promising results, the applicability, performance, and challenges of deploying foundation models for global path planning tasks remain largely unexplored for drones. Unlike previous learning-based methods focused on reactive control or environment awareness, foundation models offer the potential to perform high-level spatial reasoning and interpret multimodal prompts for trajectory generation. While existing works such as LEVIOSA [28] convert natural language commands to drone waypoints, they are limited to local inference and lack systematic evaluation across varied scenarios or integration with real-time perception.

In this work, we propose a foundation model-guided path planning framework, FM-planner, tailored specifically for autonomous drone navigation. We conduct an extensive benchmarking study that systematically evaluates eight state-of-the-art large language models (LLMs) and five vision-language models (VLMs) across diverse simulated scenarios. Leveraging insights gained from this benchmark, we further fine-tune the top-performing LLM, Llama-3.1-8B-Instruct, using trajectories collected from representative drone flights. Subsequently, we integrate this fine-tuned LLM model with a general-purpose vision encoder (LLM-Vision) to enable real-time obstacle perception and path planning (see Fig. 1). The main contributions of this research are summarized as follows:

- 1) We propose a foundation model-guided path planning framework designed specifically for autonomous drone

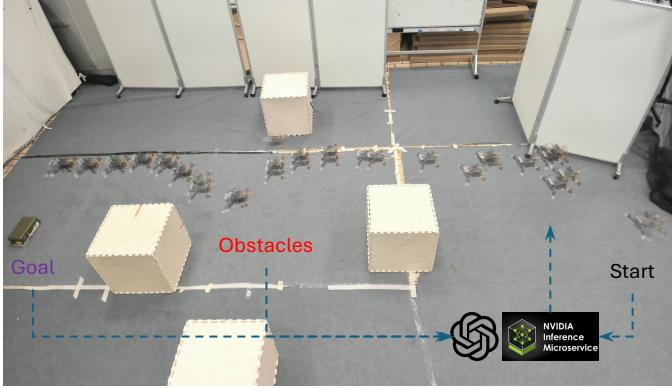


Fig. 1. The real-world autonomous flight with a foundation model-guided path planner. The obstacle positions are recognized by a vision encoder and fed into LLM with prompts.

navigation, integrating tailored prompt engineering and a multistage architecture.

- 2) To our knowledge, this work is the first to comprehensively benchmark multiple foundation models for drone path planning, systematically assessing their performance across diverse simulation scenarios.
- 3) We further deploy and validate the effectiveness of LLM-Vision-guided path planner through physical drone experiments in real-world flight scenarios.

The rest of this article is organized as follows. The related work is discussed in Section II. Section III introduces the proposed foundation model guided path planners. The experiment settings and results are presented and discussed in Section IV. Section V concludes this article.

II. RELATED WORKS

A. Path Planning

Path planning is a fundamental problem in robotics, involving the computation of optimal or feasible paths from an initial location to a destination, under constraints such as obstacle avoidance and efficient traversal. Traditional path planning methods can generally be classified into grid-based algorithms, like A* [29], and sampling-based algorithms, such as Rapidly-exploring Random Trees (RRT) and its optimal variant RRT* [30]. Grid-based methods systematically search the environment, providing guaranteed optimal solutions under specific assumptions but suffering from computational complexity in larger or higher-dimensional spaces. Sampling-based algorithms, on the other hand, scale more effectively with dimensionality, although they typically only guarantee asymptotic optimality.

Optimization-based methods have emerged to address the limitations of classical algorithms, particularly in complex environments requiring adherence to multiple objectives and constraints simultaneously. Approaches such as Fast-Planner [7] and EGO-Planner [8] employ convex optimization frameworks, demonstrating significant success in cluttered and obstacle-rich scenarios. However, they rely heavily on accurate dynamic models and computationally intensive optimization

processes, limiting their responsiveness and adaptability to highly dynamic conditions.

Learning-based approaches have gained popularity due to their capability to handle complex and unpredictable environments without explicit system modeling [31]–[33]. IL-based methods have demonstrated significant achievements in drone autonomous flight scenarios, where drones successfully imitate expert trajectories [13], [34]. DRL techniques have similarly shown potential, especially in vision-based drone navigation tasks, effectively learning control policies directly from environmental interaction [12], [35]. Despite their strengths, these methods still require large-scale, task-specific datasets for training, which restricts their generalizability and robustness in new or unseen environments.

B. Foundation Models

Foundation models, characterized by extensive pre-training on massive and diverse datasets, offer powerful generalization capabilities, substantially improving robotic perception, reasoning, and adaptability across tasks [36]. Unlike traditional task-specific models, foundation models enable rapid fine-tuning to a broad spectrum of downstream tasks, reducing the dependency on extensive scenario-specific training.

Recent applications of large language models (LLMs) have demonstrated their ability to enhance robotic task planning by integrating high-level semantic understanding and logical inference capabilities. In UAV navigation, Aikins et al. [28] leveraged LLMs to translate natural language instructions into actionable 3D waypoints, demonstrating effective trajectory planning from abstract human commands. Similarly, vision-language models (VLMs) have shown promise in directly interpreting visual information for UAV navigation and environmental reasoning [26]. Despite these advances, current research on applying foundation models to drone path planning primarily addresses reactive or local navigation tasks, leaving a significant gap in comprehensive benchmarking and validation of global path planning capabilities. Establishing such benchmarks is crucial for future developments, guiding researchers toward more robust and generalizable path planning solutions based on foundation models.

III. METHODOLOGY

This section introduces the foundation model guided path planning approach. To enable spatially informed decision-making, we design a tokenized input that combines both textual instructions and visual-perceptual information. Specifically, the prompt provided to the LLM is formulated as:

$$\mathcal{T}_{\text{input}} = [\langle \text{Prompt} \rangle, \mathbf{P}_s, \mathbf{P}_g, \{\mathbf{P}_o^n\}_{n=1}^N] \quad (1)$$

where \mathbf{P}_s and \mathbf{P}_g denote the start and goal coordinates and $\{\mathbf{P}_o^n\}$ represents the position of the n -th obstacle. This structured representation is serialized into natural language tokens and fed into the foundation model to guide global path planning. We develop the FM-planner from three stages. In Stage 1, an LLM processes provided start, goal, and obstacle coordinates to generate an initial flight plan that outlines a

coarse navigational trajectory. Stage 2 uses a dedicated VLM that focuses on obstacle detection and precise localization of start and goal positions directly. After the comparison between these 2 approaches, Stage 3 integrates the LLM with the fundamental visual encoder, providing the visual perception into the LLM-guided path planner.

A. LLM-guided Path Planner

The LLM-guided path planner (see Fig. 2 (a)) integrates the pretrained LLM model with prompt instruction, a waypoint parsing module, and a path refinement module. Multiple open-source pretrained LLMs are first evaluated for autonomous drone navigation tasks. The models, selected from the same microservice platform, include **Llama-3.1-8B-Instruct**, **Qwen-2.5-coder (7B and 32B variants)**, **Gemma-2 (9B and 27B variants)**, **Mistral (12B and 24B variants)**, and **Deepseek-R1**. The models are tested within identical environmental conditions using the NVIDIA Integration Manager (NIM) API, ensuring uniformity across model inference. The choice of diverse parameter counts for selected models allowed exploration of model scalability and performance trade-offs in spatial reasoning tasks. The few-shot prompting technique is applied to provide the LLM with extra context on the start point, goal, and the locations of the $2 \times 2 \times 2m$ cubic obstacles on the map. This enriched prompt allows the LLM to produce more spatially logical, collision-free waypoint sequences. The prompt is communicated to the LLM via NVIDIA's API, yielding an initial set of waypoints. The LLM's textual response is then parsed into numerical waypoint coordinates $\{\mathbf{P}_i\}$. Usually the generated waypoints are quite sparse. A Euclidean distance-based interpolation algorithm is adopted to smooth the path, inserting intermediate points $\{\mathbf{P}_j\}$ at $0.5m$ intervals for smoother navigation, where $\mathbf{P}_{j|i} = \mathbf{P}_i + 0.5(\mathbf{P}_{i+1} - \mathbf{P}_i)j/\|\mathbf{P}_{i+1} - \mathbf{P}_i\|$. Regarding the waypoints around the obstacles, a safety margin $1.2m$ is maintained around the obstacle along the edges. Finally, the execution module converts the refined path into flight commands by sending position setpoints to the drone via ROS.

B. VLM-guided Path Planner

The VLM-guided path planner explores drone navigation driven by pretrained VLMs, specifically targeting their ability to infer collision-free flight paths from visual environmental inputs (see Fig. 2 (b)). Five VLMs, **ChatGPT-4o**, **Gemini 2**, **Microsoft Copilot**, **Claude 3.5 Sonnet**, and **Grok 3**, were selected to evaluate the generative potential of multimodal models when given drone navigation scenarios in the form of annotated Gazebo simulation maps with starting coordinates \mathbf{P}_s and goal coordinates \mathbf{P}_g . Prompt engineering techniques are applied to extract waypoint trajectories. Using few-shot prompting, each VLM first received three example cases, each consisting of a visual map paired with an obstacle-free set of (x, y, z) waypoints. These examples guided the models in learning spatial patterns and generalizing safe trajectories. In an instruction-based prompting setup, models are given direct instructions (e.g., "Analyze the visual map and generate a safe path from start to

goal while avoiding obstacles"), leveraging their multimodal capability to translate visual context into coherent navigation plans. Each VLM returned a list of 3D waypoints representing a hypothetical collision-free trajectory from the designated start to goal positions. The raw output from the VLM is passed into the path interpolation, which applies Euclidean interpolation to smooth the motion between waypoints. A small step size of $0.1m$ is used to provide fine granularity and ensure physically feasible drone movement. This process produces a denser set of intermediate waypoints while preserving the spatial intent of the original path. The same waypoint interpolation around obstacles is adopted to smooth the path.

C. LLM-guided Path Planner with Vision Encoder

To adapt the LLM-guided path planner for practical application, the general vision encoder is integrated into the LLM module to provide the estimated positions of obstacles in real time (see Fig. 3). In this work, YOLOv8 serves as the vision encoder, efficiently processing the video feed to detect and classify obstacles and relevant scene elements. Subsequently, a pretrained LLM model, fine-tuned using Low-Rank Adaptation (LoRA) via the `Unsloth.ai` platform, interprets these visual outputs to produce real-time navigation trajectories. LoRA introduces trainable low-rank updates to the frozen pre-trained weights, drastically reducing the number of trainable parameters. LoRA enables efficient adaptation by updating only a subset of the model's parameters, specifically within the attention heads and feed-forward layers. Let $\mathbf{W}_0 \in \mathbb{R}^{d \times k}$ denote a weight matrix in the original LLM. Instead of updating \mathbf{W}_0 directly, LoRA learns a low-rank decomposition:

$$\Delta \mathbf{W} = \mathbf{B} \mathbf{A}, \quad (2)$$

where $\mathbf{B} \in \mathbb{R}^{d \times r}$ and $\mathbf{A} \in \mathbb{R}^{r \times k}$ with $r \ll \min(d, k)$. The updated forward computation becomes:

$$\mathbf{y} = \mathbf{W}_0 \mathbf{x} + \Delta \mathbf{W} \mathbf{x} = \mathbf{W}_0 \mathbf{x} + \mathbf{B}(\mathbf{A} \mathbf{x}), \quad (3)$$

where $\mathbf{x} \in \mathbb{R}^k$ is the input and $\mathbf{y} \in \mathbb{R}^d$ is the output. This approach keeps \mathbf{W}_0 frozen and only updates \mathbf{A} and \mathbf{B} , achieving parameter-efficient adaptation with $\mathcal{O}(dr + rk)$ additional parameters instead of $\mathcal{O}(dk)$. This efficiency is especially valuable for deployment on resource-constrained hardware, such as edge computers used in autonomous drones. The training objective is to minimize the cross-entropy loss between the predicted waypoint output $\hat{\mathbf{P}}$ and the ground truth waypoints \mathbf{P} over a dataset of N task samples:

$$\mathcal{L}_{CE} = -\frac{1}{N} \sum_{i=1}^N \sum_{j=1}^C \mathbf{P}_j^{(i)} \log \hat{\mathbf{P}}_j^{(i)}, \quad (4)$$

where C is the number of waypoints and $\mathbf{P}_j^{(i)}$ is the ground truth waypoints j in task $i \in N$.

Domain-specific datasets from simulation and real-world drone navigation scenarios are used to fine-tune the LLM, refining its natural language understanding and spatial reasoning for navigation tasks. Simultaneously, tailored prompts are developed with explicit drone parameters to guide the model

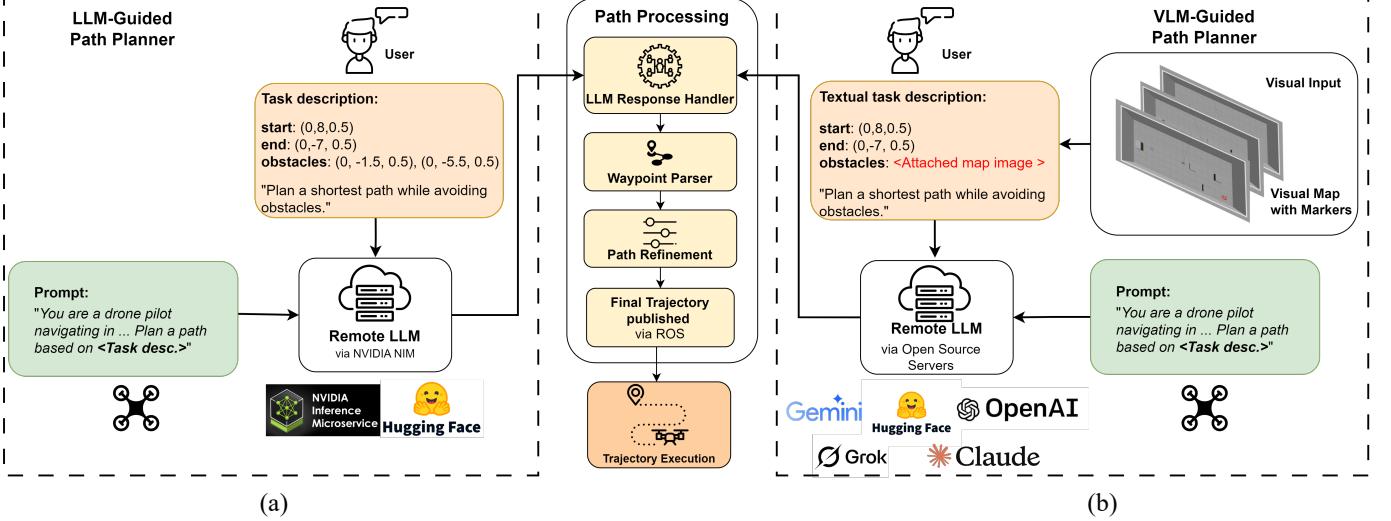


Fig. 2. The framework of the foundation model guided path planners with prompt and task descriptions. (a) The LLM-guided path planner with purely textual inputs; (b) The VLM-guided path planner with textual and visual inputs. The path processing module remains the same to extract the readable waypoint list.

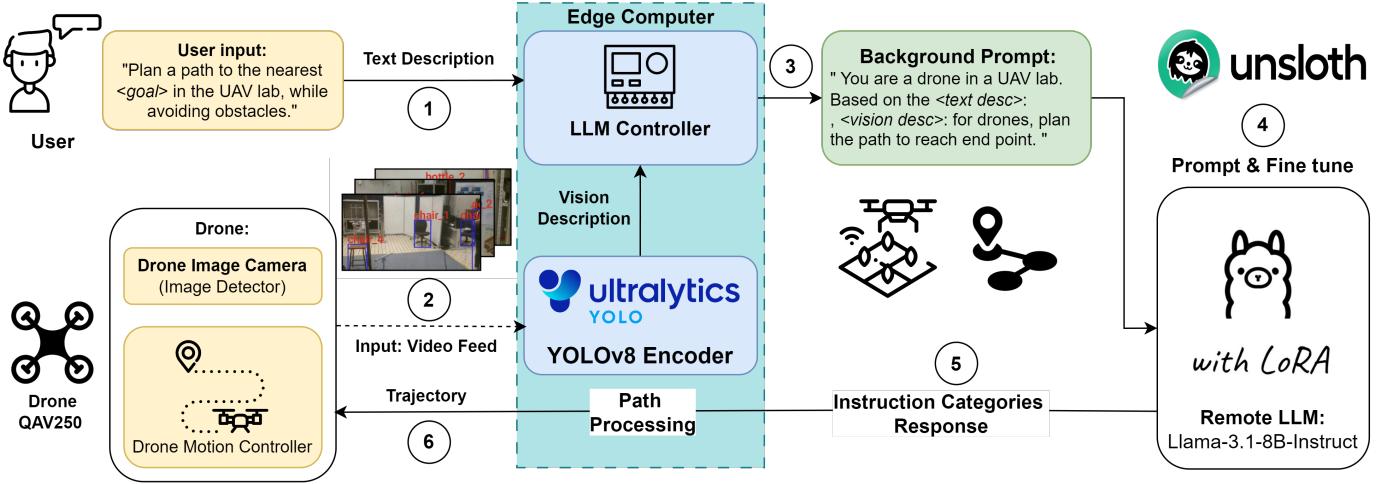


Fig. 3. System architecture of the LLM-Vision-guided path planner framework. A user-provided instruction is combined with real-time visual context from the YOLOv8 vision encoder and fed into a fine-tuned LLM. The planner generates trajectories, which are executed by the drone.

toward generating structured, accurate waypoint sequences. A synthetic dataset is generated using a custom Python script to cover a wide range of drone navigation scenarios. Each entry included coordinates for a start point, goal, and multiple obstacle positions. For each scenario, waypoints are algorithmically computed to guarantee at least a $1m$ clearance from every obstacle, ensuring collision-free navigation paths.

The vision encoder identifies environmental features, outputting structured data including class labels marked by two-dimensional (2D) bounding boxes. The depth information is further provided to estimate the 3D position. These outputs are tokenized and combined with a carefully constructed domain-specific prompt to inform navigation in complex indoor environments. The combined visual-context input is then fed into the LLM in a zero-shot prompting technique, where the model infers navigation paths without direct prior examples. Based

solely on visual labels and provided instructions, the LLM generates a sequence of waypoints in a two-dimensional plane (maintaining a fixed altitude) for the navigation trajectory.

IV. EXPERIMENTS AND RESULTS

A. Simulation Setup

These experiments aimed to evaluate how accurately and efficiently pretrained foundation models can plan autonomous drone paths using prompt instructions. Experiments were conducted within a constrained indoor Gazebo simulation environment, simulating realistic drone operational scenarios. The drone (controlled via ROS Noetic and MAVROS) navigated a 3D space populated with static cubic obstacles (size of $2m \times 2m \times 2m$). The drone is the default Iris model with the PX4 controller. The LLMs produced waypoint lists with prompts specifying start position P_s , goal position P_g , and

TABLE I
COMPARATIVE EVALUATION OF LLM MODELS OVER VARIOUS OBSTACLE AVOIDANCE SCENARIOS

Model	2 Obstacles				3 Obstacles			
	ACT (s) ↓	PL (m) ↓	SR ↑	ESS ↑	ACT (s) ↓	PL (m) ↓	SR ↑	ESS ↑
Llama-3.1-8B-Instruct	80s	14.3	100%	1.250	90s	16.1	100%	1.111
Deepseek-R1	—	—	0%	0.000	—	—	0%	0.000
Qwen-2.5-coder-7B	90s	16.1	90%	1.000	102s	18.2	100%	0.980
Qwen-2.5-coder-32B	86s	15.4	90%	1.046	87s	15.5	80%	0.920
Gemma-2-9B-it	88s	15.7	70%	0.795	90s	16.1	60%	0.667
Gemma-2-27B-it	88s	15.7	80%	0.909	96s	17.2	80%	0.833
Mistral-nemo-12B	83s	14.8	50%	0.602	105s	18.8	60%	0.571
Mistral-small-24B	83s	14.8	70%	0.843	99s	17.7	70%	0.707
A* [5]	80s	14.3	100%	1.250	87s	15.6	100%	1.115
RRT [6]	84s	15.0	50%	0.595	84s	15.0	90%	1.071
Q-learning [33]	80s	14.3	50%	0.625	87s	15.6	100%	1.115

TABLE II
DETAILED LLM MODEL COMPARISON (LLAMA-3.1-8B-INSTRUCT VS QWEN-2.5-CODER-7B)

Model	# Obs.	Coordinates of Obstacles	ACT (s) ↓	PL (m) ↓	SR ↑	ESS ↑
Llama-3.1-8B-Instruct	2	(0, -1.5, 0.5); (0, -5.5, 0.5) (0, -1.5, 0.5); (0, -5.5, 0.5);	80s	14.3	100%	1.250
	3	(-2.5, -3.5, 0.5) (0, -1.5, 0.5); (0, -5.5, 0.5);	90s	16.1	100%	1.111
	4	(-2.5, -3.5, 0.5); (-2.5, 0.5, 0.5) (0, -1.5, 0.5); (0, -5.5, 0.5); (-2.5, -3.5, 0.5); (-2.5, 0.5, 0.5);	100s	17.9	90%	0.900
	5	(-1.5, 3.5, 0.5)	110s	19.7	90%	0.818
Qwen-2.5-coder-7B	2	(0, -1.5, 0.5); (0, -5.5, 0.5) (0, -1.5, 0.5); (0, -5.5, 0.5);	90s	16.1	90%	1.000
	3	(-2.5, -3.5, 0.5) (0, -1.5, 0.5); (0, -5.5, 0.5);	102s	18.2	100%	0.980
	4	(-2.5, -3.5, 0.5); (-2.5, 0.5, 0.5) (0, -1.5, 0.5); (0, -5.5, 0.5); (-2.5, -3.5, 0.5); (-2.5, 0.5, 0.5);	110s	19.7	90%	0.818
	5	(-1.5, 3.5, 0.5)	—	0/10	0%	0.000

obstacle coordinates $\{\mathbf{P}_o\}$. The raw waypoints were then parsed and refined to ensure smooth trajectories, with intermediate waypoints interpolated using a Euclidean distance-based algorithm with a step size of $0.5m$. The simulations were running on a workstation with an RTX 3090 GPU (24 GB of G6X memory).

B. Evaluation Metrics

To evaluate the effectiveness of each pretrained language model in drone path planning, three primary metrics were used: Success Rate (SR), Average Completion Time (ACT), Path Length (PL) and the derived Efficiency-Success Score (ESS). SR is the percentage of trials where the drone reached the goal without collision, indicating reliability and obstacle avoidance capability. ACT quantifies the average time taken to complete a successful navigation task, reflecting operational efficiency and the path smoothness. Considering these metrics in isolation can be misleading; for example, a model might finish quickly but have a low success rate or achieve high success but take too long. To address this, we introduce the ESS as a composite metric, defined as $ESS = SR/ACT (\%/\text{sec})$.

C. Baseline Planners

We selected three typical path planners as the baselines, namely the A* planner [5], the RRT planner [6] and the Q-learning planner [33]. To maintain the same environment settings, we discretize the workspace $\mathcal{X} = \{(x, y) \mid x \in [-3, 3], y \in [-8, 8]\} \subset \mathbb{R}^2$ into a uniform 2D 50×50 grid. Denote the grid-cell size by $\Delta_x = \Delta_y = 0.12m$. Start and goal positions are \mathbf{P}_s and \mathbf{P}_g , and obstacle centers $\{\mathbf{P}_o\}$ are expanded into $1m \times 1m$ squares in the grid.

1) *A* Planner*: A* is a graph-search algorithm that finds an optimal path on a discretized grid by combining the actual cost from the start node with a heuristic estimate to the goal. We construct a 4-connected grid graph $G = (V, E)$, where each vertex $v \in V$ corresponds to a free cell. For an edge $(u, v) \in E$, the cost is $c(u, v) = \Delta_x = 0.12m$. Define the objective function and heuristic functions as follows:

$$g(n) = \min_{\text{path } s \rightarrow n} \sum_{(u,v)} c(u, v), \quad h(n) = |x_n - x_g| + |y_n - y_g|,$$

and the A* total cost $f(n) = g(n) + h(n)$. With admissible Manhattan heuristic $h(n)$, A* optimize the node connection

and returns an optimal path.

2) *RRT Planner*: RRT is a sampling-based planner that incrementally builds a search tree rooted at the start position by repeatedly sampling random states and extending the nearest tree node toward each sample by a fixed step size. We grow a tree T as: $x_{\text{rand}} \sim \text{Uniform}(\mathcal{X}_{\text{free}})$, $x_{\text{near}} = \arg \min_{x \in T} \|x - x_{\text{rand}}\|$, and the update rule is

$$x_{\text{new}} = x_{\text{near}} + \delta \frac{x_{\text{rand}} - x_{\text{near}}}{\|x_{\text{rand}} - x_{\text{near}}\|}, \quad (5)$$

where $\delta = 1$ cell. If $x_{\text{new}} \in \mathcal{X}_{\text{free}}$, add it to T . Stop when $\|x_{\text{new}} - \mathbf{P}_g\| \leq \epsilon$, $\epsilon = 2$ cells. RRT is probabilistically complete but not guaranteed optimal.

3) *Q-learning Planner*: We implement a simple greedy policy that, at every step, selects the neighboring grid cell minimizing Euclidean distance to the goal. At each discrete time step k , with current cell x_k , choose the next state

$$x_{k+1} = \arg \min_{x \in \mathcal{N}_4(x_k), x \notin \mathcal{O}} \|x - \mathbf{P}_g\|, \quad (6)$$

where $\mathcal{N}_4(x_k)$ are the 4-neighbors and \mathcal{O} is the obstacle set. If no neighbor reduces the distance, pick any adjacent free cell. We limit the horizon with $K_{\text{max}} = 500$ steps. This myopic policy requires no training and highlights the benefits of full DRL agents with long-horizon value estimation.

D. Benchmarking LLM-guided Path Planner

1) *Base Comparison*: We first evaluated the performance and generalizability of several LLM models and baseline planners across two distinct scenarios involving different obstacle complexities: a scenario ($S1$) with two obstacles and another scenario ($S2$) with three obstacles (see Fig. 4). Specifically, the scenario $S1$ was configured with start and goal positions at $\mathbf{P}_s = [0.0, 6.0, 0.5]$ and $\mathbf{P}_g = [0.0, -7.0, 0.5]$, respectively, with obstacle locations defined as $\{\mathbf{P}_o\} = \{[0.0, -1.5, 0.5], [0.0, -5.5, 0.5]\}$. Scenario $S2$, featuring an additional obstacle, used $\mathbf{P}_s = [0.0, 6.0, 0.5]$, $\mathbf{P}_g = [-2.5, -7.0, 0.5]$, and obstacle coordinates $\{\mathbf{P}_o\} = \{[0.0, -1.5, 0.5], [0.0, -5.5, 0.5], [-2.5, -3.5, 0.5]\}$. Their comparative performances are summarized in Table I.

In both scenarios, Llama-3.1-8B-Instruct matches the optimal A* planner on reliability and efficiency, and significantly outperforms all other neural planners. In the two-obstacle scenario ($S1$), Llama-3.1-8B-Instruct achieves a 100% success rate (SR) in average 80s over a 14.3m trajectory (ESS = 1.250). By contrast, DeepSeek-R1 fails in all 10 trials, underscoring the limitations of purely reactive methods. The mid-sized Qwen-2.5-7B variant attains a 90% SR with longer paths and lower ESS, reflecting the trade-off between distance and navigation precision.

When complexity increases with a third obstacle ($S2$), Llama-3.1-8B-Instruct again sustains 100% SR in average 90s over a 16.1m path (ESS = 1.111). Similarly, Qwen-2.5-7B attains 100% SR with ESS = 0.980, demonstrating strong generalization. In contrast, the Qwen-2.5-32B and both Gemma variants see their SR drop to approximately 80%, highlighting scalability challenges in more complex environments. The RRT planner is able to provide the shortest path but

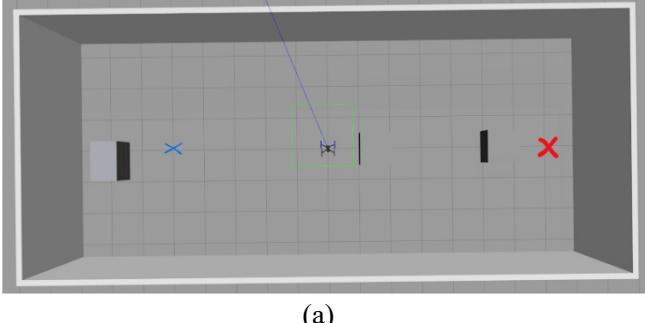
TABLE III
EFFECT OF SPEED ON NAVIGATION SUCCESS FOR
LLAMA-3.1-8B-INSTRUCT

Scenario	Rate	ACT ↓	Speed ↑	SR ↑	ESS ↑
2 Obstacles	20Hz	80s	0.179m/s	100%	1.250
	30Hz	75s	0.191m/s	100%	1.333
	40Hz	70s	0.204m/s	100%	1.429
	50Hz	60s	0.238m/s	20%	0.333
	60Hz	—	—	0%	0.000
3 Obstacles	20Hz	90s	0.179m/s	100%	1.111
	30Hz	80s	0.201m/s	100%	1.250
	40Hz	65s	0.248m/s	90%	1.385
	50Hz	60s	0.268m/s	90%	1.500
	60Hz	—	—	0%	0.000

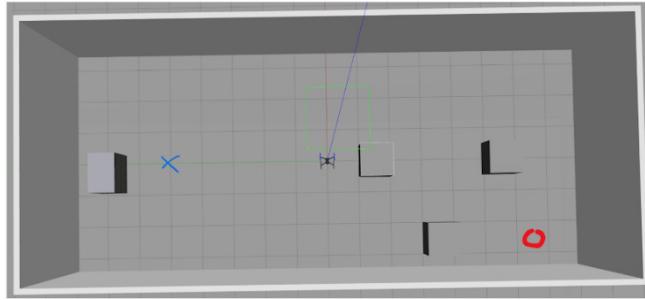
is not stable due to its random search mechanism. Given their consistently superior SR and ESS, Llama-3.1-8B-Instruct and Qwen-2.5-7B were selected for further detailed comparison.

2) *Detailed Comparison*: The detailed comparative evaluations are conducted between Llama-3.1-8B-Instruct and Qwen-2.5-coder-7B under varying levels of obstacle complexity. Specifically, the start and goal positions were consistently set to $\mathbf{P}_s = [0.0, 6.0, 0.5]$ and $\mathbf{P}_g = [-2.5, -7.0, 0.5]$, respectively. The obstacle configurations and comparative results are presented in Table II. Across all tested scenarios, Llama-3.1-8B-Instruct consistently outperformed Qwen-2.5 in terms of both reliability and navigation accuracy. While both models initially achieved high success rates with two and three obstacles, the performance of Qwen-2.5 degraded significantly as the obstacle count increased, culminating in a complete failure in scenarios with five obstacles. Meanwhile, the Qwen-2.5 generates a longer PL to ensure safety. In contrast, Llama-3.1-8B-Instruct maintained a robust success rate $\geq 90\%$ and consistently accurate navigation even in the most challenging scenario involving five obstacles. These findings highlight Llama-3.1-8B-Instruct's strong generalization capability and robustness, supporting its selection for subsequent operational speed testing.

3) *Speed Test*: To further evaluate the reliability and practical safety of the LLM-guided path planner, we assessed the performance of Llama-3.1-8B-Instruct under varying ROS publishing rates, reflecting different operational flight speeds. The start and goal positions were fixed at $\mathbf{P}_s = [0.0, 6.0, 0.5]$ and $\mathbf{P}_g = [-2.5, -7.0, 0.5]$. The environments included a two-obstacle scenario ($\mathbf{P}_o = \{[0.0, -1.5, 0.5], [0.0, -5.5, 0.5]\}$) and a three-obstacle scenario ($\mathbf{P}_o = \{[0.0, -1.5, 0.5], [0.0, -5.5, 0.5], [-2.5, -3.5, 0.5]\}$). Test outcomes are summarized in Table III. Results indicate that the Llama-3.1-8B-Instruct model reliably maintained a perfect 100% success rate at ROS publishing rates up to 40 Hz. However, performance rapidly deteriorated at 50 Hz and higher rates due to insufficient re-planning intervals between successive control updates. Consequently, a maximum practical control frequency of approximately 40 Hz is recommended, balancing responsiveness with reliable



(a)



(b)

Fig. 4. Input bird's-eye view images for the VLM-guided path planner test. (a) Two obstacles; (b) three obstacles.

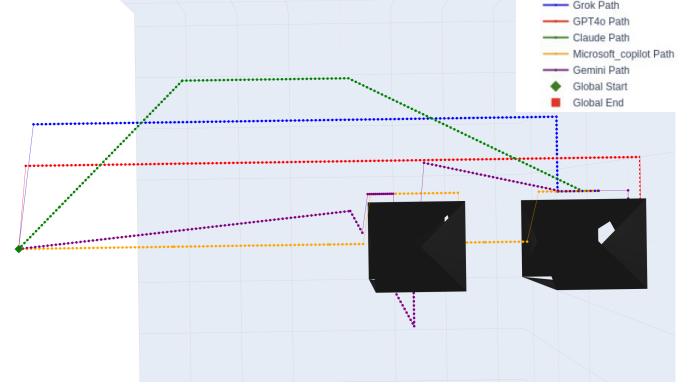
TABLE IV
LLAMA-3.1-8B-INSTRUCT FINE-TUNING PARAMETERS

Hyperparameter	Value
Batch Size	4
Epochs	60
Learning Rate	2×10^{-4}
Optimiser	AdamW (8-bit)
Activation Function	ReLU (MLP), Swish (LLM default)
Weight Decay	0.01

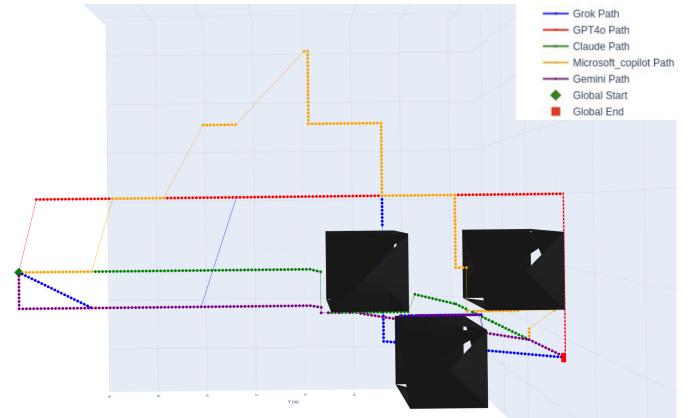
navigation performance.

E. Benchmarking VLM-guided Path Planner

1) *Settings:* This part evaluates the effectiveness of VLM-guided path planner using high-resolution visual map inputs instead of purely textual coordinate-based inputs. In contrast to the LLM-guided path planner approach, VLM-guided methods directly leverage visual context obtained from environmental imagery, emphasizing spatial reasoning grounded in visual cues. Experiments were conducted within the Gazebo ROS Noetic simulation environment, where the drone navigated from clearly defined starting positions to goal positions while avoiding cubic obstacles of size $1m \times 1m \times 1m$ placed throughout the simulated space. Precise starting and goal 3D coordinates were parsed and provided as inputs to the respective VLM models. Several representative VLMs, including ChatGPT-4o, Gemini 2, Microsoft Copilot, Claude 3.5 Sonnet, and Grok 3, were systematically evaluated to assess their potential for purely vision-based spatial reasoning. Fig. 4



(a)



(b)

Fig. 5. Paths generated from various VLMs. (a) Mission with two obstacles; (b) Mission with three obstacles. The VLMs rarely generate optimal paths.

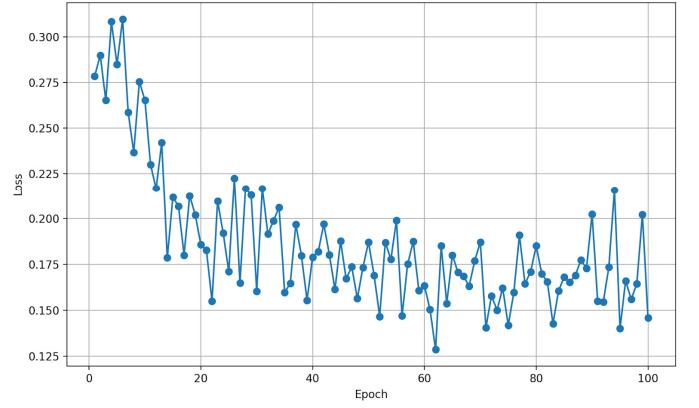


Fig. 6. The training curve of Llama-3.1-8B-Instruct during fine-tuning.

illustrates the annotated visual maps utilized as inputs for the VLMs, clearly marking the drone's initial (blue) and target (red) positions.

2) *Analysis:* Fig. 5 presents trajectories generated by all five VLMs within each scenario. GPT-4o and Claude produced relatively coherent and feasible paths, successfully avoiding obstacles, though their trajectories were often nonoptimal

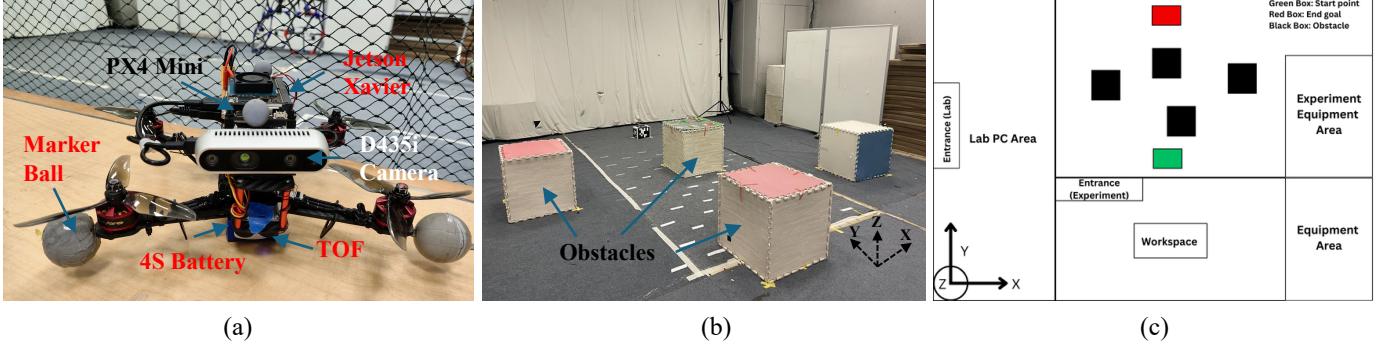


Fig. 7. The configuration of physical experiments. (a) The configuration of QAV drone with the companion computer; (b) The autonomous flight area with multiple obstacles; (c) The layout of the flight area.

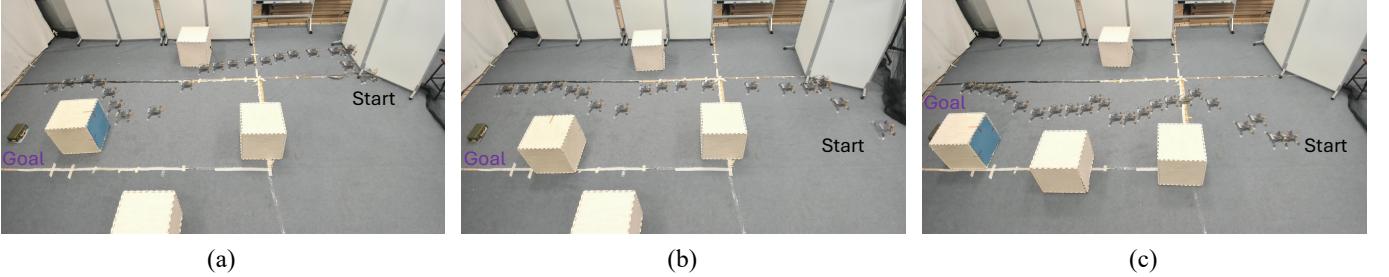


Fig. 8. The real-world autonomous flight with the developed LLM-Vision planner under various tasks in Table V. (a) Task 1; (b) Task 2; (c) Task 3.

in terms of distance or smoothness. Microsoft Copilot exhibited moderate performance, generating valid but overly conservative paths. In contrast, Gemini and Grok struggled to produce reliable trajectories, frequently exhibiting collisions or erratic behavior. Compared to the LLM-based approach, the VLM-guided planners were less consistent in generating safe and efficient paths. Despite having access to rich visual information, their performance was limited by a lack of robust spatial reasoning and insufficient generalization. On the other hand, Llama-3.1-8B-Instruct consistently delivered reliable results using only textual spatial inputs, demonstrating superior logical inference and planning capabilities. To combine the strengths of both modalities, we developed an LLM-guided path planner enhanced with a vision encoder, enabling real-time visual perception while maintaining strong spatial reasoning.

F. Fine-tuning for LLM-Vision Planner

With Llama-3.1-8B-Instruct selected as the path planner module, we further fine-tune the model with LoRA. The hyperparameters were selected based on empirical testing and prior studies on efficient LLM adaptation (see Table IV). To support this process, we curated domain-specific datasets derived from both simulated and real-world drone navigation scenarios. In parallel, we designed task-specific prompts incorporating explicit drone parameters to guide the model in generating structured and accurate waypoint sequences. A synthetic dataset comprising 5,000 entries was generated to

TABLE V
LLM-VISION PLANNER TASKS AND RESULTS

$\mathbf{P}_s/\mathbf{P}_g$	#Obs	$\{\mathbf{P}_o\}$	ACT	Success	RT
Task1: (0.5, 1.0, 0.6) (0.5, 6.3, 0.6)	4	(0.5, 2.3, 0.6) (-1.2, 3.3, 0.6) (2.5, 4.2, 0.6) (0.5, 4.4, 0.6) (0.5, 2.3, 0.6)	25s	Yes	9.6s
Task2: (-0.5, 0.5, 0.6) (0.5, 6.3, 0.6)	4	(-1.2, 3.3, 0.6) (2.5, 4.2, 0.6) (0.0, 4.4, 0.6) (0.0, 2.3, 0.6)	23s	Yes	9.8s
Task3: (-0.5, 1.0, 0.6) (0.5, 6.3, 0.6)	4	(-0.5, 3.3, 0.6) (2.5, 4.2, 0.6) (0.0, 5.0, 0.6)	20s	Yes	9.4s

represent diverse navigation scenarios. Each entry includes coordinates for a start point \mathbf{P}_s , a goal point \mathbf{P}_g , and multiple obstacle positions $\langle \mathbf{P}_o^i \rangle$. Since the RRT planner provides optimal shortest paths, waypoints were computed using the RRT planner to maintain a minimum 1m clearance from all obstacles, ensuring safe and collision-free trajectories.

The model was trained over 100 epochs, and the training performance was monitored using cross-entropy loss, which quantifies how closely the model's predicted distributions align with the ground truth. As shown in Fig. 6, the training loss converged to approximately 0.15 after 60 epochs, indicating stable and effective fine-tuning.

G. Physical Experiments

To validate the effectiveness of the proposed LLM-Vision-guided path planner, we deployed the fine-tuned model on a QAV250 racing drone equipped with a Jetson Xavier NX edge computer. Physical experiments were conducted in an indoor flight space of size $8 \times 10 \times 5m$, as illustrated in Fig. 7. The drone operated in offboard mode, receiving self-localization data streamed at 120 Hz from an OptiTrack motion capture system. The start and goal positions were manually specified by the user, while obstacle information was extracted from RGB and depth images using the onboard vision encoder. When the LLM-Vision-guided planner was employed, real-time perception was provided by the RealSense D435i RGBD camera. The captured images were resized and processed with the vision encoder before being passed to the Llama-3.1-8B-Instruct model, which was requested from the Hugging Face server and executed at each planning step (see Fig. 3, Step 2).

We evaluated the planner across multiple scenarios with varying start positions and obstacle configurations. The test setup and performance results are summarized in Table V, and example flight trajectories are shown in Fig. 8. More results are available in the supplementary video. Across all tested conditions, the LLM-Vision-guided planner successfully generated obstacle-avoiding trajectories, guiding the drone to its targeted goal. As illustrated in Fig. 8 (c), when obstacles were placed closely along the direction to the goal, the planner was still able to generate aggressive waypoints along the obstacles. The model's average reasoning time (RT) was approximately 9 seconds per query, which is acceptable for real-time, onboard autonomous flight applications since the global path is not required to update at high frequency.

V. CONCLUSION

In this work, we introduced a novel foundation model-guided framework named FM-planner for drone path planning, systematically benchmarking multiple LLM and VLM approaches. Our extensive evaluation revealed that pretrained foundation models, particularly VLMs, lack global spatial reasoning capabilities compared to LLMs without specific textual descriptions. In particular, fine-tuned LLMs integrated with vision encoders demonstrate robust spatial reasoning and real-time obstacle awareness, making them well-suited for practical drone global planning tasks. Physical experiments confirmed the feasibility and reliability of our approach in real-world autonomous flight. Future work includes exploring advanced multimodal foundation models for complex, dynamic environments and further optimizing real-time computational performance for broader operational scenarios.

REFERENCES

- [1] J. Xiao, P. Pisutsin, and M. Feroskhan, “Collaborative target search with a visual drone swarm: An adaptive curriculum embedded multistage reinforcement learning approach,” *IEEE Transactions on Neural Networks and Learning Systems*, 2023.
- [2] P. M. Kornatowski, M. Feroskhan, W. J. Stewart, and D. Floreano, “Downside up: Rethinking parcel position for aerial delivery,” *IEEE Robotics and Automation Letters*, vol. 5, no. 3, pp. 4297–4304, 2020.
- [3] K. Liu and B. M. Chen, “Industrial uav-based unsupervised domain adaptive crack recognitions: From database towards real-site infrastructural inspections,” *IEEE Transactions on Industrial Electronics*, vol. 70, no. 9, pp. 9410–9420, 2023.
- [4] Q. Jia, J. Xiao, and M. Feroskhan, “Multitarget assignment under uncertain information through decision support systems,” *IEEE Transactions on Industrial Informatics*, vol. 20, no. 8, pp. 10 636–10 646, 2024.
- [5] Q. Zhou and G. Liu, “Uav path planning based on the combination of a-star algorithm and rrt-star algorithm,” in *2022 IEEE International Conference on Unmanned Systems (ICUS)*, 2022, pp. 146–151.
- [6] B. Hu, Z. Cao, and M. Zhou, “An efficient rrt-based framework for planning short and smooth wheeled robot motion under kinodynamic constraints,” *IEEE Transactions on Industrial Electronics*, vol. 68, no. 4, pp. 3292–3302, 2021.
- [7] B. Zhou, F. Gao, L. Wang, C. Liu, and S. Shen, “Robust and efficient quadrotor trajectory generation for fast autonomous flight,” *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3529–3536, 2019.
- [8] X. Zhou, Z. Wang, H. Ye, C. Xu, and F. Gao, “Ego-planner: An esdf-free gradient-based local planner for quadrotors,” *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 478–485, 2021.
- [9] J. Xiao, R. Zhang, Y. Zhang, and M. Feroskhan, “Vision-based learning for drones: A survey,” *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–21, 2025.
- [10] Q. Sun, J. Fang, W. X. Zheng, and Y. Tang, “Aggressive quadrotor flight using curiosity-driven reinforcement learning,” *IEEE Transactions on Industrial Electronics*, vol. 69, no. 12, pp. 13 838–13 848, 2022.
- [11] J. Xiao and M. Feroskhan, “Learning multipursuit evasion for safe targeted navigation of drones,” *IEEE Transactions on Artificial Intelligence*, vol. 5, no. 12, pp. 6210–6224, 2024.
- [12] Y. Zhang, J. Xiao, and M. Feroskhan, “Learning cross-modal visuomotor policies for autonomous drone navigation,” *IEEE Robotics and Automation Letters*, vol. 10, no. 6, pp. 5425–5432, 2025.
- [13] E. Kaufmann, L. Bauersfeld, A. Loquercio, M. Müller, V. Koltun, and D. Scaramuzza, “Champion-level drone racing using deep reinforcement learning,” *Nature*, vol. 620, no. 7976, pp. 982–987, 2023.
- [14] L. Chen, Y. Wang, Y. Mo, Z. Miao, H. Wang, M. Feng, and S. Wang, “Multiagent path finding using deep reinforcement learning coupled with hot supervision contrastive loss,” *IEEE Transactions on Industrial Electronics*, vol. 70, no. 7, pp. 7032–7040, 2023.
- [15] L. Zhang, W. Yi, H. Lin, J. Peng, and P. Gao, “An efficient reinforcement learning-based cooperative navigation algorithm for multiple uavs in complex environments,” *IEEE Transactions on Industrial Informatics*, vol. 20, no. 10, pp. 12 396–12 406, 2024.
- [16] J. Xiao, P. Pisutsin, and M. Feroskhan, “Toward collaborative multitarget search and navigation with attention-enhanced local observation,” *Advanced Intelligent Systems*, p. 2300761, 2024.
- [17] C. Yan, C. Wang, X. Xiang, Z. Lan, and Y. Jiang, “Deep reinforcement learning of collision-free flocking policies for multiple fixed-wing uavs using local situation maps,” *IEEE Transactions on Industrial Informatics*, vol. 18, no. 2, pp. 1260–1270, 2022.
- [18] J. Xiao, X. Fang, Q. Jia, and M. Feroskhan, “Learning resilient formation control of drones with graph attention network,” *IEEE Internet of Things Journal*, pp. 1–1, 2025.
- [19] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat *et al.*, “Gpt-4 technical report,” *arXiv preprint arXiv:2303.08774*, 2023.
- [20] J.-B. Alayrac, J. Donahue, P. Luc, A. Miech, I. Barr, Y. Hasson, K. Lenc, A. Mensch, K. Millican, M. Reynolds *et al.*, “Flamingo: a visual language model for few-shot learning,” *Advances in neural information processing systems*, vol. 35, pp. 23 716–23 736, 2022.
- [21] F. Lin, X. Li, W. Lei, J. J. Rodriguez-Andina, J. M. Guerrero, C. Wen, X. Zhang, and H. Ma, “Pe-gpt: A new paradigm for power electronics design,” *IEEE Transactions on Industrial Electronics*, vol. 72, no. 4, pp. 3778–3791, 2025.
- [22] S. Qin, Y. Zhao, H. Wu, L. Zhang, and Q. He, “Harnessing the power of large language model for effective web api recommendation,” *IEEE Transactions on Industrial Informatics*, pp. 1–11, 2025.
- [23] H. Wang, C. Li, and Y.-F. Li, “Large-scale visual language model boosted by contrast domain adaptation for intelligent industrial visual monitoring,” *IEEE Transactions on Industrial Informatics*, vol. 20, no. 12, pp. 14 114–14 123, 2024.
- [24] L. Xia, Y. Hu, J. Pang, X. Zhang, and C. Liu, “Leveraging large language models to empower bayesian networks for reliable human-robot collaborative disassembly sequence planning in remanufacturing,” *IEEE Transactions on Industrial Informatics*, vol. 21, no. 4, pp. 3117–3126, 2025.

- [25] Y. Cui, S. Niekum, A. Gupta, V. Kumar, and A. Rajeswaran, “Can foundation models perform zero-shot task specification for robot manipulation?” in *Learning for dynamics and control conference*. PMLR, 2022, pp. 893–905.
- [26] X. Wang, D. Yang, Z. Wang, H. Kwan, J. Chen, W. Wu, H. Li, Y. Liao, and S. Liu, “Towards realistic uav vision-language navigation: Platform, benchmark, and methodology,” *arXiv preprint arXiv:2410.07087*, 2024.
- [27] R. Firooz, J. Tucker, S. Tian, A. Majumdar, J. Sun, W. Liu, Y. Zhu, S. Song, A. Kapoor, K. Hausman *et al.*, “Foundation models in robotics: Applications, challenges, and the future,” *The International Journal of Robotics Research*, p. 02783649241281508, 2023.
- [28] G. Aikins, M. P. Dao, K. J. Moukpe, T. C. Eskridge, and K.-D. Nguyen, “Leviosa: Natural language-based uncrewed aerial vehicle trajectory generation,” *Electronics*, vol. 13, no. 22, p. 4508, 2024.
- [29] P. E. Hart, N. J. Nilsson, and B. Raphael, “A formal basis for the heuristic determination of minimum cost paths,” *IEEE transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968.
- [30] S. Karaman and E. Frazzoli, “Sampling-based algorithms for optimal motion planning,” *The international journal of robotics research*, vol. 30, no. 7, pp. 846–894, 2011.
- [31] X. Hai, Z. Zhu, Y. Liu, A. W. H. Khong, and C. Wen, “Resilient real-time decision-making for autonomous mobile robot path planning in complex dynamic environments,” *IEEE Transactions on Industrial Electronics*, pp. 1–12, 2025.
- [32] E. Kaufmann, A. Loquercio, R. Ranftl, M. Müller, V. Koltun, and D. Scaramuzza, “Deep drone acrobatics,” *Proceedings of Robotics: Science and Systems XVI*, 2020.
- [33] J. Wu, Y. Sun, D. Li, J. Shi, X. Li, L. Gao, L. Yu, G. Han, and J. Wu, “An adaptive conversion speed q-learning algorithm for search and rescue uav path planning in unknown environments,” *IEEE Transactions on Vehicular Technology*, vol. 72, no. 12, pp. 15 391–15 404, 2023.
- [34] B. Yang, H. Shi, and X. Xia, “Federated imitation learning for uav swarm coordination in urban traffic monitoring,” *IEEE Transactions on Industrial Informatics*, vol. 19, no. 4, pp. 6037–6046, 2023.
- [35] Y. Zhang, C. Yan, J. Xiao, and M. Feroskhan, “Npe-drl: Enhancing perception constrained obstacle avoidance with non-expert policy guided reinforcement learning,” *IEEE Transactions on Artificial Intelligence*, 2024.
- [36] R. Bommasani, D. A. Hudson, E. Adeli, R. Altman, and et al., “On the opportunities and risks of foundation models,” *arXiv preprint arXiv:2108.07258*, 2021.