Next-Generation LLM for UAV:
From Natural Language to Autonomous Flight

Liangqi Yuan*, *Student Member, IEEE,* Chuhao Deng*, *Student Member, IEEE,* Dong-Jun Han, *Member, IEEE,*
Inseok Hwang, *Member, IEEE,* Sabine Brunswicker, and Christopher G. Brinton, *Senior Member, IEEE*

Abstract—With the rapid advancement of Large Language Models (LLMs), their capabilities in various automation domains, particularly Unmanned Aerial Vehicle (UAV) operations, have garnered increasing attention. Current research remains predominantly constrained to small-scale UAV applications, with most studies focusing on isolated components such as path planning for toy drones, while lacking comprehensive investigation of medium- and long-range UAV systems in real-world operational contexts. Larger UAV platforms introduce distinct challenges, including stringent requirements for airport-based take-off and landing procedures, adherence to complex regulatory frameworks, and specialized operational capabilities with elevated mission expectations. This position paper presents the Next-Generation LLM for UAV (NELV) system—a comprehensive demonstration and automation roadmap for integrating LLMs into multi-scale UAV operations. The NELV system processes natural language instructions to orchestrate short-, medium-, and long-range UAV missions through five key technical components: (i) LLM-as-Parser for instruction interpretation, (ii) Route Planner for Points of Interest (POI) determination, (iii) Path Planner for waypoint generation, (iv) Control Platform for executable trajectory implementation, and (v) UAV monitoring. We demonstrate the system's feasibility through three representative use cases spanning different operational scales: multi-UAV patrol, multi-POI delivery, and multi-hop relocation. Beyond the current implementation, we establish a five-level automation taxonomy that charts the evolution from current LLM-as-Parser capabilities (Level 1) to fully autonomous LLM-as-Autopilot systems (Level 5), identifying technical prerequisites and research challenges at each stage. Project page with code and videos: https://liangqiyuan.github.io/NeLV/.

Index Terms—Large Language Model, Unmanned Aerial Vehicle, Planning, Autonomous Systems

Fig. 1. Overview of the NELV System.

I. INTRODUCTION

The rise of Large Language Models (LLMs) has transformed numerous domains, such as mobile services, vehicles, and robotics [1]–[3]. These fields have become increasingly intelligent and user-friendly through LLM integration, enabling command and control through natural language. This conversation-based control between humans and LLMs improves both the 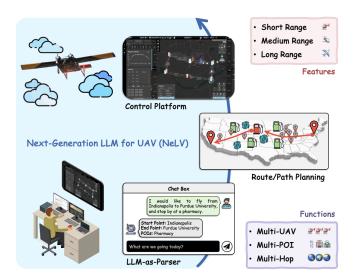ability of LLMs to interpret context and the convenience with which humans can direct LLMs to perform actions [4]. LLMs fulfill diverse roles within these systems. LLM-as-Router can orchestrate task allocation and model selection for human pilots, LLM-as-Agent can execute actions on behalf of humans, and LLM-as-Judge can conduct evaluations in place of human judgment. The increasing specialization of LLMs in distinct roles throughout systems optimizes their performance through clear responsibility allocation. Consequently, LLMs are becoming an essential component of next generation autonomous control and self-driving technologies.

Unmanned Aerial Vehicles (UAVs) represent an important component of next generation transportation [17], [18]. According to data from the Federal Aviation Administration (FAA), registered UAVs in the United States exceed 1 million as of March 2025, with 427,335 remote pilots certified [19]. As UAV numbers increase, this human pilot workforce proves inadequate and represents one of the limitations preventing UAVs from assuming a more substantial role in transportation. How to enable fewer human pilots to operate a single UAV or allow one human pilot to control multiple UAVs remains an ongoing research question. This challenge necessitates assistance from powerful automation systems to help human pilots process information, comprehend data, suggest operations, execute actions, and communicate effectively. LLMs emerge as strong candidates not only because they accept language input, which proves considerably more efficient than

* Equal contribution

L. Yuan and C. G. Brinton are with the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907, USA. E-mail: liangqiy@purdue.edu; cgb@purdue.edu

C. Deng and I. Hwang are with the School of Aeronautics and Astronautics, Purdue University, West Lafayette, IN 47907, USA. E-mail: deng113@purdue.edu; ihwang@purdue.edu

D.-J. Han is with the Department of Computer Science and Engineering, Yonsei University, Seoul, South Korea. E-mail: djh@yonsei.ac.kr

S. Brunswicker is with the Polytechnic Institute, Purdue University, West Lafayette, IN 47907, USA. E-mail: sbrunswi@purdue.edu

TABLE I
COMPARISON OF RELATED LITERATURE OF LLM FOR UAV SYSTEM

Literature	Year	Type	Key Topic	Implementation	Case Study		
					Short-Range	Medium-Range	Long-Range
[5]	2024	Survey	LLMs for UAVs	-	-	-	-
[6]	2025	Survey	LLMs for UAVs	-	-	-	-
[7]	2024	Survey	Generative AI for UAVs	-	-	-	-
[8]	2024	Survey	Generative AI for UAV Networks	-	-	-	-
[9]	2024	Survey	Generative AI for UAV Networks	-	-	-	-
[10]	2024	Benchmark	Simulation Platform	Simulator	City UAV Simulator	-	-
[11]	2023	Research	LLM for UAV Planning	Indoor Small UAV	Mission Planning	-	-
[12]	2023	Research	LLM for UAV Planning	Simulator	Mission Planning	-	-
[13]	2024	Research	LLM for UAV Planning	Indoor Small UAV	Object Detection	-	-
[14]	2024	Research	LLM for UAV Planning	Indoor Small UAV	Obstacle Prediction	-	-
[15]	2024	Research	LLM for UAV Planning	Indoor Small UAV	Mission Planning	-	-
[16]	2024	Research	LLM for UAV Planning	Outdoor Small UAV Swarm	Formation	-	-
Ours	2025	Position Paper	• LLM-as-Parser • Route Planning • Path Planning • Control Platform • UAV Monitoring	Outdoor Medium UAV	Multi-UAV Patrol	Multi-POI Delivery	Multi-Hop Relocation

alternative command inputs, but also because their robust reasoning capabilities enable them to process UAV-related information effectively.

Although considerable literature focuses on LLM for UAV systems, most studies remain confined to singular aspects, such as using LLMs solely for small toy UAV planning, as detailed in Table I. A significant research gap exists regarding a comprehensive system that progresses from human language input to determining potential path nodes, developing detailed paths, integrating with UAV platforms, and ultimately controlling UAVs. Most of the literature exclusively addresses the planning of small UAVs without considering medium- and long-range UAV planning. These categories exhibit substantial differences. Small UAVs, typically multirotors, can take off from any location and perform short-distance missions, but possess limited cargo capacity and lower performance capabilities. Medium- and long-range UAVs, typically fixed-wings, require airport infrastructure and the adherence to strict take-off and landing procedures. They must also comply with FAA regulations, including prohibitions against flying over crowds and requirements to avoid air traffic control zones. These considerations present new and unique challenges for LLM for UAV systems.

In this paper, we present the Next-Generation LLM for UAV (NELV) system, illustrated in Figure 1. This LLM-powered UAV system translates human language input into autonomous control of short-, medium-, and long-range UAVs that perform various missions. The system incorporates multiple key technical components, including LLM-as-Parser, route planning, path planning, and control platform. Human pilots can access the input for each component and intervene to control or modify operations. While we envision future systems where LLMs play central roles in planning, control, and decision-making, current LLM capabilities remain limited for safety-critical UAV operations. Therefore, this position paper serves dual purposes: (1) *System Demonstration:* We present a working implementation where LLMs handle natural language interpretation (Level 1 of our taxonomy) while traditional algorithms perform route planning, path planning, and control; and (2) *Automation Roadmap:* We establish a five-level taxonomy that defines technical requirements, identifies key challenges, and charts development pathways from current LLM-as-Parser capabilities to future LLM-as-Autopilot systems capable of fully autonomous decision-making.

The contributions of this position paper are as follows.

- **End-to-End System Framework and Implementation:** We introduce NELV, the first complete framework integrating LLMs with multi-scale UAV operations from natural language input to executable flight trajectories. To our knowledge, NELV represents the first LLM-powered UAV system capable of supporting short-, medium-, and long-range operations, as well as complex missions involving multi-UAV patrol, multi-POI delivery, and multi-hop relocation scenarios.
- **Five-Stage Pipeline Architecture:** We develop a comprehensive pipeline encompassing: (i) *LLM-as-Parser* that engages in conversational interaction with human pilots to iteratively refine flight plans and interpret operational preferences; (ii) *Route Planner* that considers pilot-specified constraints and multi-objective optimization criteria; (iii) *Path Planner* that circumvents restricted zones, including controlled airspace and adverse meteorological conditions; (iv) *Control Platform* that generates executable trajectories incorporating airport-specific take-off and landing patterns; and (v) *UAV Monitor* that provides real-time mission execution with safety pilot intervention capabilities.

- **Multi-Scale Mission Demonstration:** We demonstrate the system's capabilities through three representative use cases spanning different operational scales: (UC1) short-range multi-UAV patrol, wherein five UAVs conduct surveillance of forested areas within a 5 km radius; (UC2) medium-range multi-POI delivery with two delivery objectives; and (UC3) long-range multi-hop relocation from New York to Los Angeles with intermediate refueling stops. The system integrates real-time airspace information, population density data, and weather risk assessments to enable safe autonomous operations at any airport within the continental United States.
- **Five-Level Automation Roadmap:** We establish a systematic taxonomy defining the evolution from current capabilities to fully autonomous systems: (L1) *LLM-as-Parser* for instruction interpretation, (L2) *LLM-as-Route-Planner* for strategic route optimization, (L3) *LLM-as-Path-Planner* for tactical path determination and collision avoidance, (L4) *LLM-as-Executor* for integrated control system coordination, and (L5) *LLM-as-Autopilot* for fully autonomous flight operations. For each level, we identify specific technical prerequisites, required knowledge bases, and key research challenges.

The remainder of this paper is organized as follows. Section II reviews the existing literature on LLM for UAV systems. Section III introduces the overall NELV system architecture and the five key technical components. In Section IV, we present the experimental configuration, datasets utilized, and three representative use cases that demonstrate system capabilities. Before concluding the paper in Section VI, we discuss five hierarchical levels of future research directions and development pathways in Section V.

II. RELATED WORK

A. LLM for UAV

A significant characteristic of LLMs compared to traditional machine learning methods is their powerful reasoning and generalization capabilities that autonomously complete various tasks. This is particularly important for UAVs, including context understanding, perception, planning, decision-making, and more [20], [21]. Additionally, LLMs enable more natural language-based communication with human pilots, reducing operational burden [22] and potentially allowing non-expert users to operate UAVs in the future. Unlike other automated tasks such as autonomous driving, UAVs are categorized as short-range, medium-range, and long-range vehicles, which can be further classified into fixed-wing, multi-rotor, and other configurations [6]. Each type requires different operational approaches. UAV operation presents greater challenges than that of autonomous vehicles due to various factors. These diverse configurations and structures pose unique challenges in the deployment and generalization of LLMs in the UAV domain. This diversity in configurations, structures, and operational methods also results in a lack of large UAV datasets for LLM training, which is a fundamental issue hindering the deep development of LLMs in this field. Currently, most datasets are limited to bird's-eye view detection of ground

objects such as people, vehicles, roads, fires, buildings, and more [10], [23]–[26], representing only a minimal portion of UAV missions. Related datasets for UAV control, planning, and human-pilot language dialogue are scarce [27], [28].

Therefore, our proposed NELV system aims to address these research gaps. We present a complete system workflow for LLM for UAV, from language input to actual UAV flight control. Unlike related literature that focuses on complex computer vision tasks (such as identifying ground objects), we emphasize UAV planning, control, and flight. Notably, the NELV system can incorporate any state-of-the-art vision model to accomplish various tasks, making our system complementary to other vision models. Most importantly, we demonstrate that the NELV system can operate short-range, medium-range, and long-range UAVs. This provides readers with a reliable platform for developing various UAV configurations.

B. UAV Autonomous System

Flight plan development in UAV autonomous systems constitutes a prerequisite for mission execution. It encompasses five fundamental phases: take-off, en-route flight, mission execution, return flight, and landing [29]. This structure proves particularly essential for fixed-wing UAVs and large-scale UAVs, as deviation from these established protocols potentially precipitates severe safety incidents [30]. The existing UAV literature predominantly focuses on small toy UAVs, particularly multirotor configurations, or neglects UAV traffic patterns [31], [32]. Furthermore, UAV missions in the literature are predetermined by human pilots without addressing real-world flight planning challenges involving ambiguous objectives and numerous alternatives. For example, for refueling operations, pilots encounter multiple airport options with varying fuel costs, where human pilots select options based on costs, quality, safety, or other factors. Subsequently, after selecting the appropriate nodes, the specific path planning requires a reasonable 3D planning based on multiple objectives [33]–[38], presenting a unique challenge that must simultaneously accommodate FAA regulations while determining optimal paths from infinite possible waypoints. Moreover, the trajectory builds upon beyond path planning by incorporating traffic patterns for take-off, landing, and mission execution [39], requiring an appropriate configuration according to airport specifications. Recall the objectives of our NELV system, all three aforementioned procedural steps necessitate consideration of various UAV types, flight ranges, missions, and additional variables, introducing significant implementation challenges.

The proposed NELV system aims to incorporate and deploy various UAV configurations to accomplish diverse missions. The planning problem progresses hierarchically from high-level node selection to specific path determination, and ultimately to comprehensive trajectory specification. Our NELV system extends the relevant literature by considering how to determine nodes in UAV missions, such as selecting from multiple candidate mission locations. This approach not only enhances mission flexibility, facilitating more adaptable and
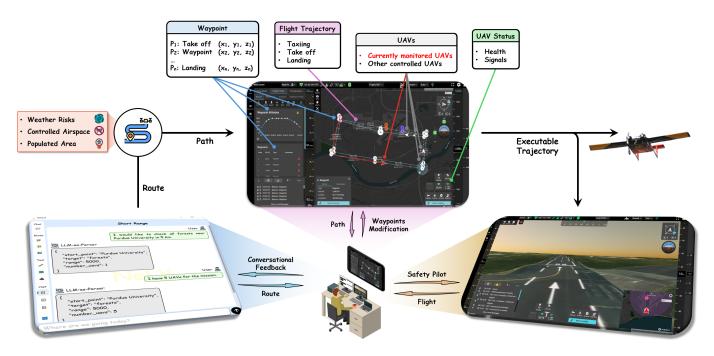
Fig. 2. Operational Framework and User Interface of the NELV System. **Left – Chat Box:** The pilot interacts with the LLM through natural-language instructions. After receiving an initial flight plan, the pilot performs route and path planning with visualization support. **Top – Control Platform [40]:** Once the path is uploaded, the pilot can further refine the mission by manually adjusting waypoints. **Right – UAV Monitor:** The safety pilot continuously monitors UAVs' status, position, and flight behavior during operation and can make real-time adjustments as needed.

highly customized flight operations, but also substantially reduces cognitive demands on human pilots through preference-based adaptive flight planning protocols. Furthermore, our NELV system's comprehensive design architecture thoroughly addresses all flight components, including take-off, landing, and mission-specific traffic patterns, providing practical solutions for real-world applications.

III. NEXT-GENERATION LLM FOR UAV SYSTEM

A. NELV *System Overview*

The NELV system interprets missions through natural language input from human pilots, performs route planning based on flight specifications, executes path planning according to environmental constraints and airspace restrictions, implements executable trajectories through the control platform, and finally monitors UAV flight operations, as illustrated in Figure 2. The current NELV system comprises five key components: (i) *LLM-as-Parser* for interpreting human pilot instructions, (ii) *Route Planner* for determining nodes along the flight trajectory, (iii) *Path Planner* for establishing specific paths between each pair of nodes, (iv) *Control Platform* for generating executable trajectories, and (v) *UAV Monitoring* for ultimate flight execution. Pilots retain the capability to intervene and modify system outputs at any stage following component execution through the three interfaces depicted in Figure 2, enabling real-time adaptation to suboptimal planning results or evolving mission requirements.

Mathematically, the NELV system operates by constructing a UAV flight graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{X}, \mathcal{Y})$, where $\mathcal{V}$ represents the set of nodes, $\mathcal{E}$ represents the set of edges, $\mathcal{X}$ is the set of node attributes, and $\mathcal{Y}$ denotes the node types.

(i) *LLM-as-Parser* determines nodes $\mathcal{E}$ along with their attributes $\mathcal{X}$ and types $\mathcal{Y}$ based on the UAV mission described by the human pilot, such as forest wildfire surveillance or supermarket supply delivery operations. A specific forest area or supermarket establishment constitutes a node $v \in \mathcal{V}$. These nodes possess distinct attributes $x \in \mathcal{X}$, including geographical coordinates (latitude/longitude), operational status, and congestion levels. The node type $y \in \mathcal{Y}$ represents the designated flight pattern and operational mode, as aircraft may execute aerial cargo drops for supermarket deliveries while airports facilitate conventional landing and take-off procedures.

(ii) *Route Planner* performs preliminary route planning based on $\mathcal{G}$, where a route is a sequence of nodes that can be expressed as

$$\xi = [v_1, v_2, \ldots, v_N], \tag{1}$$

where $v_i \in \mathcal{V}$ and $N \geq 2$ is an integer representing the number of nodes in the route. The route ξ must include take-off and landing nodes, which may be identical. Route planning here is multi-objective, always considering the balance between mission completion, flight quality, and cost.

(iii) *Path Planner* based on the preliminary route ξ generates a more detailed and specific path:

$$\pi = [v_1, p_{1,2}^{(1)}, p_{1,2}^{(2)}, \ldots, v_2, p_{2,3}^{(1)}, p_{2,3}^{(2)}, \ldots, v_N], \tag{2}$$

where $p_{i,i+1}^{(j)}$ represents the j-th waypoint in the path segment connecting node v_i to node v_{i+1}. This detailed path planning incorporates comprehensive FAA regulations governing various airspace classifications.

Algorithm 1: NELV System

```
/* LLM-as-Parser */
```
Input: Pilot instructions
Output: Graph $\mathcal{G}$ and constraints $\mathbf{C}^{\max}$
Execute: Algorithm 2
Interface: Chat Box $\triangleright$ Fig. 2 - Left

```
/* Route Planner */
```
Input: Graph $\mathcal{G}$ and constraints $\mathbf{C}^{\max}$
Output: Route ξ $\triangleright$ Eq. (1)
Execute: Algorithm 3
Interface: Chat Box $\triangleright$ Fig. 2 - Left

```
/* Path Planner */
```
Input: Route ξ $\triangleright$ Eq. (1)
Output: Path π $\triangleright$ Eq. (2)
Execute: Algorithm 4
Interface: Chat Box $\triangleright$ Fig. 2 - Top

```
/* Control Platform */
```
Input: Path π $\triangleright$ Eq. (2)
Output: Trajectory ρ $\triangleright$ Eq. (3)
Execute: Algorithm 5
Interface: Control Platform $\triangleright$ Fig. 2 - Top

```
/* UAV Monitor */
```
Input: Trajectory ρ $\triangleright$ Eq. (3)
Execute: UAV
Interface: UAV Monitor $\triangleright$ Fig. 2 - Right

(iv) *Control Platform* transforms the path π into an executable trajectory:

$$\rho = [\mathbf{V}_1, p_{1,2}^{(1)}, p_{1,2}^{(2)}, \ldots, \mathbf{V}_2, p_{2,3}^{(1)}, p_{2,3}^{(2)}, \ldots, \mathbf{V}_N], \quad (3)$$

where each $\mathbf{V}_i$ represents a specialized air traffic pattern corresponding to the node type $y_i \in \mathcal{Y}$ of node v_i. These patterns include specific procedures such as take-off sequences, loiter points, and landing procedures tailored to the operational context of each node.

(v) *UAV Execution and Monitoring:* The UAV receives trajectory coordinates and associated control commands from the control platform. The onboard flight control systems autonomously navigate the aircraft to sequential waypoints while maintaining prescribed flight parameters. Concurrently, the UAV's integrated sensor suite transmits real-time telemetry data, including precise geolocation coordinates, fuel levels, engine RPM, oil temperature, and other critical operational parameters, to the ground control platform. Remote operators continuously monitor all system parameters and maintain supervisory control authority to implement corrective interventions when operational anomalies or safety considerations necessitate manual override.

Next, we will introduce the algorithms and functions of each key technical component.

B. LLM-as-Parser

LLM-as-Parser constitutes the most fundamental component of the NELV system, as it defines the UAV flight graph $\mathcal{G}$, and its accuracy directly impacts the outcomes of all subsequent processes. Although we envision future

---

Algorithm 2: LLM-as-Parser and Graph Construction

Input from Human Pilot: Initial instruction (I_0) and subsequent instructions (I_t for $t > 0$)
Input from Map Service: Node attributes ($\mathcal{X}$)
Output: Graph ($\mathcal{G}$) and constraints $\mathbf{C}^{\max}$
1   Initialize sets of nodes $\mathcal{V}$, types $\mathcal{Y}$, and constraints $\mathbf{C}^{\max}$
2   Initialize conversation $\mathcal{I} \leftarrow I_0$
3   Initialize time index $t \leftarrow 0$
4   **while** *instruction I_t exists* **do**
5   Concatenate instruction to history $\mathcal{I} \leftarrow \mathcal{I} \oplus I_t$
6   LLM reason using conversation $R_t \leftarrow \text{LLM}(\mathcal{I})$
7   Extract information from response: $\mathcal{Y}, \mathbf{C}^{\max} \leftarrow R_t$
8   Search from Map Service to obtain nodes $\mathcal{V}$ and their attributes $\mathcal{X}$ based on node types $\mathcal{Y}$
9   Concatenate response to history $\mathcal{I} \leftarrow \mathcal{I} \oplus R_t$
10      $t \leftarrow t + 1$
11   Initialize set of edges $\mathcal{E}$
12   Obtain number of nodes $N \leftarrow |\mathcal{V}|$
13   **for** $i = 1, \ldots, N$ **do**
14   **for** $j = 1, \ldots, N$ **do**
15   **if** $i \neq j$ **then**
16   Compute edge weight $w_{i,j}$ between nodes v_i and v_j
17   Add edge $\mathcal{E} \leftarrow \mathcal{E} \cup \{(v_i, v_j, w_{i,j})\}$
18   Construct graph $\mathcal{G} \leftarrow (\mathcal{V}, \mathcal{E}, \mathcal{X}, \mathcal{Y})$

Algorithm 3: Route Planner

Input from LLM-as-Parser: Graph ($\mathcal{G}$) and constraints ($\mathbf{C}^{\max}$)
Output: Optimal route (ξ^*)
1   Initialize candidate route set Ω
2   **for** *each set $\mathbf{y} \subseteq \mathcal{Y}$* **do**
3   **for** *permutation $\vec{\mathbf{y}}$ of $\mathbf{y}$* **do**
4   **if** $\vec{\mathbf{y}}$ *satisfies constraints* $\mathbf{C}^{\max}$ **then**
5   Construct a subgraph $\mathcal{G}_{\vec{\mathbf{y}}}$ containing only nodes of types in $\vec{\mathbf{y}}$
6   Search subgraph $\mathcal{G}_{\vec{\mathbf{y}}}$ to find locally optimal route $\widetilde{\xi}$ $\triangleright$ Eq. (4)
7   Add $\widetilde{\xi}$ to candidate set: $\Omega \leftarrow \Omega \cup \{\widetilde{\xi}\}$
8   **if** $\Omega \neq \emptyset$ **then**
9   Select optimal route ξ^* from Ω based on human pilot or objective function $\mathcal{O}_{\text{route}}(\xi)$ $\triangleright$ Eq. (4)
10   Early stop
11   **else**
12   Output direct route if no feasible solution $\xi^* \leftarrow [v_1, v_N]$

LLMs possessing capabilities beyond merely defining graph $\mathcal{G}$ to encompass route determination ξ, path planning π, and executable trajectory generation ρ, we have determined that current general-purpose LLMs (e.g., GPT-4o [41]) lack sufficient capability due to the absence of specialized LLMs trained on UAV planning datasets. Therefore, we presently utilize LLMs exclusively as instruction parsers to extract essential information from natural language instructions, including departure points, destination coordinates, and mission specifications.

In a typical operational scenario, a human pilot might instruct: "I would like to check all forests near Purdue University within 5 km." The LLM-as-Parser systematically extracts

"Purdue University" as the reference point, "forests" as the surveillance target, and "5000" as the operational radius. This parsing process becomes challenging when accounting for inherent variability in human linguistic behavior, including potential typographical errors, linguistic ambiguities, and incomplete information specifications. Human pilots must supplement or correct previous instructions through iterative dialogue, particularly when mission parameters require modification due to operational factors, such as alterations in departure and arrival points or temporal constraints imposed by air traffic control clearances. Continuing the aforementioned scenario, if the pilot initially omits the number of UAVs required for mission execution, they can subsequently provide clarification by stating: "I have 5 UAVs for the mission." Such iterative refinements exemplify the dynamic nature of mission planning, where initial instructions may be incomplete or require adjustment based on evolving operational requirements. These modifications can occur at any stage within the NELV system pipeline, including immediate corrections during initial instruction input, post-route planning adjustments when selecting optimal cost or time-efficient alternatives, or post-path planning modifications necessitated by adverse meteorological conditions. Consequently, a critical functionality of LLM-as-Parser involves processing human pilot interactions through conversational interfaces to dynamically modify flight plans while maintaining mission coherence.

C. Route Planning

Following the construction of the graph $\mathcal{G}$ by LLM-as-Parser, the next step involves identifying a route within $\mathcal{G}$ from the start point v_1 to the end point v_N that fulfills the mission requirements. These key points are represented as nodes $v \in \mathcal{V}$ in $\mathcal{G}$. The fundamental goal of route planning is to determine a sequence of nodes as the route $\xi = [v_1, v_2, \ldots, v_N]$ based on multiple objectives, such as identifying optimal fueling locations according to fuel costs and consumption profiles, while accommodating various constraints, such as time limitations. Mathematically, this optimization problem can be formulated as:

$$\xi^* = \underset{\xi \in \Omega}{\arg\min} \ \mathcal{O}_{\text{route}}(\xi)$$

$$\mathcal{O}_{\text{route}}(\xi) = \sum_{i=1}^{n-1} \Big(\alpha \cdot \mathcal{L}(v_i, v_{i+1}) - (1-\alpha) \cdot \mathcal{R}(v_{i+1}) \Big),$$

$$\text{s.t. } \mathbf{C}(\xi) \leq \mathbf{C}^{\max}$$

$$M(\xi) = 1$$

$$(4)$$

where ξ^* represents the optimal route, Ω denotes the feasible route space from start point to end point, and $\mathcal{O}_{\text{route}}(\pi)$ characterizes the objective function governing the route optimization process. $\mathcal{L}(v_i, v_{i+1})$ represents the cost function between consecutive nodes, incorporating factors such as air traffic density, prevailing wind conditions, fuel consumption, etc. $\mathcal{R}(v_{i+1})$ denotes the reward or utility associated with visiting node v_{i+1}. The parameter α balances the trade-off between minimizing costs and maximizing rewards. $\mathbf{C}(\xi)$ represents a vector of constraint functions with upper bound

---

Algorithm 4: Path Planner

Input from Route Planner: Route (ξ)
Input from Services: Service data ($\mathcal{S}$)
Output: Optimal path (π^*)

1 Initialize population $\mathcal{P}$, personal best positions $\mathbf{p}_0$, global best $\mathbf{g}_0$, and velocity $\mathbf{v}_0$
2 **for** $t = 1, \ldots, T-1$ **do**
3   Generate random vectors $\mathbf{r}_p$ and $\mathbf{r}_g$
4   Update velocity and positions   ▷ Eq. (6)
5   Evaluate objective function $\mathcal{O}_{path}(\mathcal{P}_{t+1})$ ▷ Eq. (5)
6   **for** $i = 2, \ldots, N$ **do**
7   **if** $\mathcal{O}_{path}(\mathcal{P}_{t+1}[i]) < \mathcal{O}_{path}(\mathbf{p}_t[i])$ **then**
8   Update personal best: $\mathbf{p}_{t+1}[z] \leftarrow \mathcal{P}_{t+1}[z]$
9   **else**
10   Maintain personal best: $\mathbf{p}_{t+1}[z] \leftarrow \mathbf{p}_t[z]$
11   Update global best: $\mathbf{g}_{t+1} \leftarrow \arg\min_{p \in \mathbf{p}_{t+1}} \mathcal{O}_{path}(p)$
12 Output optimal path $\pi^* \leftarrow \mathbf{g}_T$

$\mathbf{C}^{\max}$, encompassing various operational limitations. $M(\xi)$ signifies the mission completion indicator function, which is equal to 1 if all mission requirements are satisfied and 0 otherwise.

These functions are determined by UAV specifications, mission parameters, and operational contexts, and are configured on a case-by-case basis according to specific requirements. For instance, when executing a long-range multi-hop relocation mission requiring intermediate refueling, the cost function $\mathcal{L}(v_i, v_{i+1})$ incorporates fuel pricing variations across different airports and flight-related consumption metrics, while the reward function $\mathcal{R}(v_{i+1})$ may assume negligible values when fuel quality considerations are not operationally relevant. The constraint function $\mathbf{C}(\xi)$ represents current fuel capacity levels and range limitations, while the mission completion criterion $M(\xi)$ indicates successful refueling operations and mission objective fulfillment. Comprehensive descriptions of each function implementation for the respective use cases are provided in Section IV.

D. Path Planning

Following the determination of the optimal route ξ^* by the route planner, the path planner generates a more detailed and specific path π, accounting for a comprehensive set of service data $\mathcal{S}$, which encompasses weather forecasts, ground risks, airspace restrictions, and various regulatory parameters to ensure compliance with FAA regulations while minimizing operational costs. Given the nodes from ξ, the path planning problem involves finding intermediate waypoints $p_{i,i+1}^{(j)} \in \mathbb{R}^3$ (representing longitude, latitude, and altitude) to construct a path π that optimizes a composite objective function. For each segment connecting nodes v_i and v_{i+1}, the path planning problem can be formulated as:

$$\pi_{i,i+1}^* = \underset{\pi \in \Pi_{i,i+1}}{\arg\min} \ \mathcal{O}_{\text{path}}(\pi)$$

$$\mathcal{O}_{\text{path}}(\pi) = \beta_c \cdot \Phi(\pi) + \beta_f \cdot \mathcal{F}(\pi) \qquad (5)$$

where $\pi_{i,i+1}^*$ designates the optimal path segment between nodes v_i and v_{i+1}, $\Pi_{i,i+1}$ is the set of all feasible paths

---

Algorithm 5: Take-off and Landing Pattern Generator

Input: latitude-longitude coordinates (φ_s, λ_s), start altitude (h_s), runway heading (θ_0), waypoint separation (d), altitude profile (h), traffic pattern (TP)

Output: Local Circuit (**V**)

1 Traffic pattern sign $s \leftarrow \begin{cases} -1, & \text{TP = Left} \\ +1, & \text{TP = Right} \end{cases}$;

2 Fixed angular offsets $\Theta_{RA} \leftarrow s \cdot 90°$, $\Theta_{IA} \leftarrow s \cdot 12°$

3 Heading-increment $\boldsymbol{\Delta\theta} \leftarrow$
   $[0, 0, \Theta_{RA}, \Theta_{RA}, \Theta_{RA}, \Theta_{IA}, -2\Theta_{IA}, \Theta_{RA}, \Theta_{RA}, 0]$

4 $\mathbf{V} \leftarrow \{(\varphi_s, \lambda_s, h_0)\}$

5 **for** $k = 1, \ldots, 9$ **do**

6 $\quad \theta_k \leftarrow \theta_{k-1} + \Delta\theta_k$

7 $\quad (\varphi_k, \lambda_k) \leftarrow \mathcal{P}(\varphi_{k-1}, \lambda_{k-1}; d_k, \theta_k)$ $\qquad \triangleright$ Eq. (8)

8 $\quad \mathbf{V} \leftarrow \mathbf{V} \cup \{(\varphi_k, \lambda_k, h_k)\}$
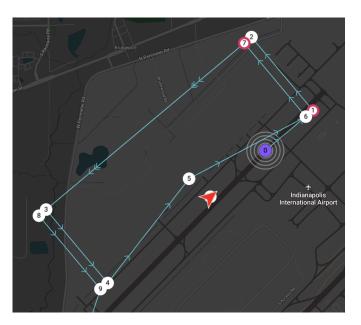
9 **return V**

Fig. 3. Standard take-off circuit pattern configured in a kidney-shaped trajectory to enable the UAV to maintain controlled loitering within the traffic pattern until air traffic control (ATC) clearance is received. The purple waypoint (0) indicates the initial take-off position and white waypoints represent intermediate navigation points along the flight path. The blue connecting lines illustrate the planned trajectory segments between consecutive waypoints, with the directional arrows indicating the prescribed flight direction and sequence. Waypoint 5 is strategically positioned to prevent the aircraft from overflying the active runway, ensuring compliance with aviation safety protocols and airport operational procedures.

between these nodes, and $\mathcal{O}_{\text{path}}(\pi)$ constitutes the objective function of path planning evaluated with respect to the service data $\mathcal{S}$. β_c and β_f are the coefficients for $\Phi(\pi)$ and $\{(\pi)$, respectively. $\Phi(\pi)$ quantifies constraint violations along the path, and $\mathcal{F}(\pi)$ represents the cost function evaluated on this path segment. β_c is much larger than β_f to ensure constraints are satisfied. Unlike route planning, which has finite solutions with a limited number of nodes, path planning presents infinite potential solutions with numerous possible waypoints. Consequently, constraints are incorporated directly into the objective function rather than treated separately as in Eq. (4). Similarly, path planning is also highly case-specific, depending on UAV type, size, weight, and mission scenario. Thus, both the cost function $\mathcal{F}(\pi)$ and the constraint function $\Phi(\pi)$ vary according to specific requirements. As an illustrative example, the cost function $\mathcal{F}(\pi)$ typically incorporates multiple factors relevant to UAV operations, including path length, mission duration, weather-related hazards, and ground-associated risks. Weather risks are typically derived from metrics such as Simplified Forecast Icing Potential (SFIP), Convective Available Potential Energy (CAPE), and the Bulk Richardson Number (BRN), which characterize icing conditions, turbulence, and thunderstorm potential. Ground risks are computed on the basis of the population density in the overflight areas and the trajectory length through these regions. Additionally, wind information can be leveraged to optimize mission time for reduced fuel consumption.

The complete optimal path π^* is constructed by concatenating the optimal path segments $\pi^*_{i,i+1}$ between consecutive nodes in route ξ, yielding $\pi^* = v_1^* \oplus \pi^*_{1,2} \oplus v_2^* \oplus \pi^*_{2,3} \oplus \cdots \oplus \pi^*_{n-1,n} \oplus v_N^*$, where $\oplus$ denotes path concatenation. To effectively and efficiently determine π within a large search space comprising both discrete and continuous variables, we employ Particle Swarm Optimization (PSO) [42], a meta-heuristic algorithm well suited for this class of problems. For path planning applications, PSO begins by randomly generating a set of candidate paths, collectively forming an initial population $\mathcal{P}_0$. This population evolves iteratively for

T generations according to the following update equations:

$$\begin{aligned} \mathbf{v}_{t+1} &= w\mathbf{v}_t + c_1 \mathbf{r}_p \cdot (\mathbf{p}_t - \mathcal{P}_t) + c_2 \mathbf{r}_g \cdot (\mathbf{g}_t - \mathcal{P}_t) \\ \mathcal{P}_{t+1} &= \mathcal{P}_t + \mathbf{v}_{t+1} \end{aligned}, \quad (6)$$

where $t = 1, 2, \ldots, T$ denotes the generation index. The parameters w, c_1, and c_2 represent inertia weight, personal influence, and social influence, respectively, which are hyperparameters for tuning PSO. The term $\mathbf{p}_t$ represents the personal best positions historically achieved by each particle and $\mathbf{g}_t$ denotes the global best solution identified across all particles at generation t. The vectors $\mathbf{r}_p$ and $\mathbf{r}_g$ contain uniformly distributed random numbers in the interval $[0, 1]$. The detailed algorithmic implementation of this PSO-based path planner is presented in Algorithm 4.

E. Control Platform

To transform the optimal path π^* into a realistic mission profile suitable for UAV deployment, the initial waypoint v_1 and terminal waypoint v_N must be replaced with standardized take-off and landing circuit patterns, respectively. Based on the coordinates of v_1 and v_N, the corresponding airports are identified through geospatial matching algorithms. Subsequently, the coordinates and headings of all runway endpoints are retrieved from the AirNav database [43] and utilized to compute the runway centerpoint (φ_s, λ_s), which serves as the reference origin for circuit generation. Given predefined altitude specifications and waypoint separation parameters, a local circuit pattern is systematically generated

using Algorithm 5, where the waypoint sequence is indexed by $k = 0, 1, \ldots, 8$. The geodetic calculations for circuit waypoint generation begin with the angular distance normalization, where d_k represents the great-circle distance between consecutive waypoints k and $k + 1$ in meters:

$$\delta_k = \frac{d_k}{6.371 \times 10^6} \tag{7}$$

where δ_k represents the angular distance normalized by Earth's radius (6.371×10^6 meters). Using this normalized angular distance δ and the bearing angle θ_k from waypoint $k-1$ to waypoint k, the latitude and longitude coordinates of each waypoint are computed through spherical trigonometry:

$$\varphi_k = \arcsin(\sin \varphi_{k-1} \cos \delta_k + \cos \varphi_{k-1} \sin \delta_k \cos \theta_k)$$
$$\lambda_k = \lambda_{k-1} + \operatorname{atan2}(\sin \theta_k \sin \delta_k \cos \varphi_{k-1}, \tag{8}$$
$$\cos \delta_k - \sin \varphi_{k-1} \sin \varphi_k)$$

where φ_k denotes the latitude and λ_k denotes the longitude of the k-th waypoint. An exemplary circuit configuration made with [40] demonstrating these geodetic transformations is illustrated in Figure 3. Finally, mission-specific operational commands, such as clockwise loitering maneuvers around designated waypoints for specified durations to facilitate cargo drop-off or loading operations, are systematically appended to construct the final executable trajectory ρ, which is subsequently uploaded to the control platform for autonomous mission execution.

IV. IMPLEMENTATION AND CASE STUDY

A. Graphical User Interface and Setup

We use PyQT5 [44] to develop a graphical user interface (GUI) that functions as the Chat Box, enabling the human pilot to interact with the LLM. The interface integrates both route planning and path planning functionalities, as illustrated in Figures 4a, 5a, and 6a. The GUI consists of three main components: Toolbar, Message Panel, and Input Box. The **Toolbar** allows the pilot to switch between different LLM modes. From top to bottom, the available options are: Chat (Speech Balloon Emoji), Short Range (Helicopter Emoji), Medium Range (Small Airplane Emoji), Long Range (Airplane Emoji), Plan Route (Pencil Emoji), Plan Path (World Map Emoji), Upload Path (Joystick Emoji), Hist - Short Range (Clipboard Emoji), Medium Range (Clipboard Emoji), Long Range (Clipboard Emoji), and Reset (Counterclockwise Arrows Button Emoji). Pilots employ Chat for general-purpose queries, while the three task modes enable planning for different missions. At any point during the interaction, pilots can activate Plan Route, Plan Path, or Upload Path to perform route planning, path planning, or upload the path to the control platform. In addition, the three Hist functions store past conversations and provide sample tasks. Finally, pilots can reset the conversation to initiate a new mission.

When pilots enter text into the **Input Box** and click the Send button (Upward Arrow), the text is combined with a predefined system prompt and submitted to the LLM for inference. In our implementation, we employ Phi-4-mini-Instruct [45] as the LLM-as-Parser. The system stores conversational context so that pilots can iteratively refine the flight plan. The **Message Panel** displays all responses, including the LLM's outputs as well as generated routes and paths. Upon receiving pilot commands, the LLM interprets the input and produces an initial text-based flight plan. Once the pilot accepts this plan, clicking Plan Route triggers the system to generate a route, which is then displayed in the Message Panel. If the route is accepted, the pilot proceeds by clicking Plan Path, prompting the system to produce the corresponding path. Finally, once the path is approved, the pilot clicks Upload Path to send it to the control platform, after which subsequent operations can be performed within the control environment.

B. Control Platform and Real UAV Configurations

We utilize the Windracers ULTRA UAV [46] in conjunction with its integrated control platform, comprising Windracers Mission Control and Windracers Autopilot [40], to execute flight control and operational missions. Mission Control constitutes a sophisticated multi-UAV distributed control architecture that provides real-time telemetry feedback regarding aircraft operational status. The platform enables pilots to dynamically modify or reconfigure mission parameters during flight operations; for instance, pilots can manually delete waypoints, prompting autonomous navigation to subsequent destinations, or adjust waypoint coordinates (latitude, longitude, and altitude) through either interactive map manipulation or direct numerical parameter editing. Mission Control additionally supports advanced operational capabilities, including loiter point configurations that enable aircraft to maintain loitering patterns over designated surveillance areas. Furthermore, the platform facilitates simultaneous visualization and control of multiple UAV assets, enabling coordinated monitoring and collaborative mission execution. While Mission Control and Autopilot platform provide superior integration and seamless control with the ULTRA UAV, NELV can alternatively be implemented using open-source autopilot software platforms such as PX4 [47] and ArduPilot [48].

The ULTRA [46] is a fixed-wing UAV with a maximum take-off mass of 510 kg and a useful payload capacity of 150 kg. The recently updated ULTRA MK2 extends this capability by transporting up to 80 liters of cargo over ranges of over 1,000 km. Powered by two HIRTH F23 engines, each rated at 45 horsepower, and fueled by a standard unleaded petrol/oil mixture, the aircraft achieves a cruise speed of approximately 40 m/s and operates at altitudes up to 4,000 m. Depending on configuration, it sustains autonomous flights for 7–9 hours and is compatible with diverse runway conditions, including paved, grass, gravel, and dirt surfaces. Beyond its physical specifications, the ULTRA platform supports a range of mission profiles: cargo delivery, parachute drops for critical supplies, and aerial survey and detection applications such as wildfire monitoring.
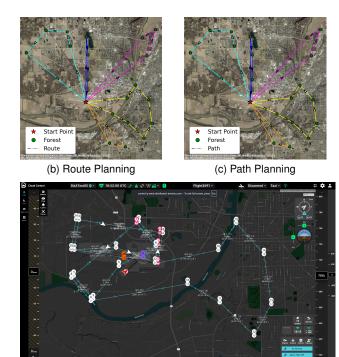
C. Use Case Setup

We next present three representative use cases of varying complexity, corresponding to short-, medium-, and long-range missions. Furthermore, we examine three specific operational

(a) LLM-as-Parser

(b) Route Planning

(c) Path Planning

(d) Executable Trajectory

Fig. 4. Use Case 1: Short-Range Multi-UAV Patrol (Sec. IV-D).

challenges: multi-UAV patrol, multi-POI delivery, and multi-hop relocation, which are evaluated across three distinct datasets and sources, including OpenStreetMap [49], the Yelp Open Dataset [50], and USA airport fuel prices and specifications from AirNav [43]. We aim to demonstrate the scalability of the NELV system through these diverse data sources, which provide geospatial information through varied methodologies and formats. For instance, OpenStreetMap serves as a comprehensive map search service enabling users to query information for any global location, while the Yelp Open Dataset constitutes a preprocessed repository containing extensive POI geographical coordinates and user review data. Additionally, users can deploy the NELV system on proprietary datasets or commercial mapping services such as Google Maps, demonstrating our system's adaptability across different data infrastructures.

In our experimental implementation, we incorporate airspace information, population density data, and weather forecasts for path planning operations. Airspace information is obtained from OpenAIP [51], a worldwide aeronautical database containing controlled airspace boundaries, flight restriction zones, and air traffic control sectors. The system evaluates airspace violations by checking flight path intersections with restricted geometric zones and applies constraint penalties for unauthorized airspace penetration. Population density [49] considerations are implemented through ground risk assessment, where the system evaluates flight paths against city boundaries and urban area geometries to minimize risks over densely populated regions, though this constraint is disabled for long-range flights operating at high altitudes. Weather forecast data is sourced from the

High-Resolution Rapid Refresh (HRRR) model stored in the Herbie dataset [52], providing meteorological predictions at different pressure altitudes (250 mb for long-range flights and 950 mb for short-range operations). The weather analysis incorporates atmospheric parameters including cloud mixing ratios, temperature, relative humidity, vertical velocity, CAPE, wind shear components, and horizontal wind vectors. These parameters are processed to generate composite risk indices, including SFIP and BRN, which are integrated along flight path segments and weighted by distance to assess cumulative weather hazards for path optimization.

D. Use Case 1: Short-Range Multi-UAV Patrol

Implementation Details: We utilize OpenStreetMap [49] as the primary source of geospatial information. OpenStreetMap constitutes a freely accessible, open-source mapping service that enables pilots to query geographical elements through keyword-based searches, including amenity classifications (e.g., education, healthcare, transportation), building categories (e.g., sports, automotive, storage), commercial establishments (e.g., food, beauty, clothing), and various other geographic features. The platform supports multiple query methodologies for identifying objects within target regions, encompassing bounding box queries, administrative boundary searches, distance-based range queries, and polygonal area selections. In our implementation, we establish the map center at Purdue University Airport and examine forested areas within a 5 km radius to facilitate a multi-UAV patrol mission focused on wildfire detection.

Figure 4a demonstrates the interactive dialogue between the pilot and the LLM, where initial mission parameters are

(a) LLM-as-Parser

(b) Route Planning
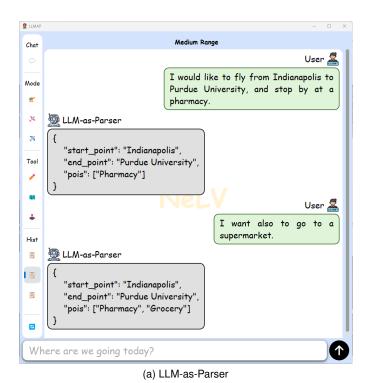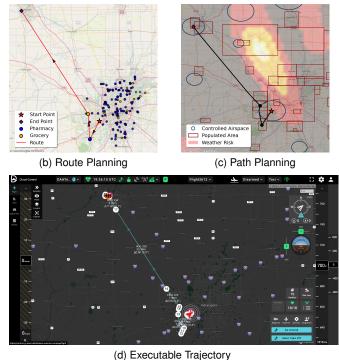

(c) Path Planning

(d) Executable Trajectory

Fig. 5. Use Case 2: Medium-Range Multi-POI Delivery (Sec. IV-E).

specified, including the origin point, the detection objective, the surveillance range, and the size of the UAV fleet. In the initial dialogue exchange, the LLM successfully identifies the mission parameters; however, given the absence of explicit UAV quantity specification, the system defaults to a single-UAV configuration. Through the utilization of conversational LLMs, NELV enables pilots to iteratively refine mission specifications through natural language interaction. For instance, following the pilot's specification of five UAVs, the LLM dynamically updates the mission plan accordingly. Upon pilot confirmation of the flight plan, the route planning algorithm is initiated. For multi-UAV route optimization, we employ OR-Tools [53], with the resulting routes illustrated in Figure 4b. The forested terrain adjacent to Purdue University is partitioned into five discrete sectors, each assigned a designated UAV route. The corresponding flight paths are depicted in Figure 4c; notably, in this short-range operational scenario, the paths coincide with the routes due to the limited operational range and altitude constraints, which render weather, airspace, and terrain-based risks negligible. Figure 4d illustrates the complete executable trajectory, including take-off and landing patterns, as well as flight paths to execute the mission. We demonstrate two distinct operational patterns utilizing different runway orientations: the take-off pattern aligned in a northeast-southwest direction and landing oriented in a nearly east-west configuration. These differentiated runway approaches prevent runway incursions when multiple UAVs execute concurrent missions at the same airport facility.

E. Use Case 2: Medium-Range Multi-POI Delivery

Implementation Details: We utilize the Yelp Open Dataset [50] as the primary source of POI information for mission planning, encompassing geographical coordinates, establishment ratings, review counts, and operational hours. The Yelp Open Dataset constitutes limited mapping data, as its repository contains POI information exclusively for 11 metropolitan areas, lacks comprehensive coverage, and does not provide real-time updates. In contrast, while OpenStreetMap services provide POI geographical coordinates, they do not incorporate user-generated metrics such as ratings and review counts. An alternative commercial mapping solution is Google Maps, which offers more comprehensive information with real-time updates; however, each query incurs associated service fees. In our implementation, we consider a use case wherein a UAV departs from Indianapolis Airport and navigates to Purdue University Airport, with intermediate stops at a pharmacy and a supermarket for supply procurement. The mission objective is to maximize the quality of these two locations (i.e., ratings and review counts) while simultaneously minimizing the total route distance. At each POI, the system automatically establishes a loiter point where the UAV executes circular flight patterns at approximately 300 meters Above Ground Level (AGL) to simulate cargo operations. In practical deployment scenarios, this configuration can be dynamically adapted based on operational requirements, such as executing precision landings for ground-based cargo pickup, performing aerial cargo drops through automated release mechanisms, or maintaining extended surveillance patterns for reconnaissance missions.

In Figure 5a, the pilot specifies a flight trajectory from

(a) LLM-as-Parser

(b) Route Planning

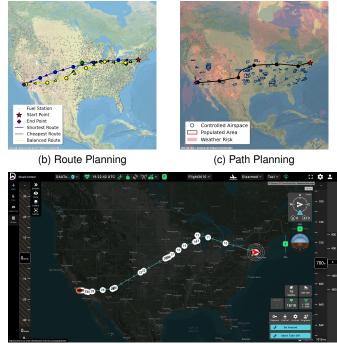(c) Path Planning

(d) Executable Trajectory

Fig. 6. Use Case 3: Long-Range Multi-Hop Relocation (Sec. IV-F).

Indianapolis to Purdue University with an intermediate stop at a pharmacy. The NELV system accurately recognizes the start and end points while correctly classifying the pharmacy as a POI. During this use case, the pilot dynamically adds an additional POI by requesting a visit to a supermarket. Following the LLM's acknowledgment and the pilot's confirmation of mission parameters, the pilot utilizes the interface tools to execute Algorithm 3 for route optimization, with results presented in Figure 5b. Subsequently, Figure 5c illustrates the path transformation from the computed route, which incorporates meteorological hazards, terrain-based risks, and airspace restrictions. In Figure 5c, the red boundaries delineate municipal and township limits, while the blue circles represent controlled airspace requiring air traffic control clearance. Additionally, a superimposed heatmap visualizes weather-related risk distributions across the operational area. The path planning algorithm enables weight adjustment to prioritize specific cost functions over others; in this particular use case, we configure the system to prioritize path distance optimization over other considerations such as weather avoidance and airspace restrictions.

F. Use Case 3: Long-Range Multi-Hop Relocation

Implementation Details: We utilize airport data obtained from AirNav [43], encompassing locations, identifiers, and fuel pricing for all airports across the United States. Fuel prices exhibit substantial variation among different airport locations. Based on data acquired from AirNav in March 2025, the continental United States contains 2,577 airports, of which 2,076 provide fuel pricing information. Aviation fuel prices exhibit substantial variation across locations, ranging

from a minimum of 0.74 USD/L at Clintonville Municipal Airport to a maximum of 2.77 USD/L at Kodiak Airport. Consequently, when UAVs conduct long-range multi-hop relocation missions, the trade-off between fuel costs at different airports and flight distances must be carefully considered. In our configuration, the ULTRA UAV features a fuel tank capacity of 80 L with a fuel consumption rate of 10.95 km/L, yielding a maximum flight range of approximately 876 km. Note that fuel consumption varies with mass and weather conditions. Due to airport traffic control constraints and varying operational complexity, estimating the actual fuel consumption during take-off and landing procedures at individual airports presents significant challenges. Therefore, we establish a standardized fuel overhead of approximately 10 liters per airport operation (take-off and landing combined), equivalent to approximately one hour of cruise flight consumption. This establishes a critical path planning constraint wherein edges between airports cannot exceed 876 km, enabling the construction of a graph network for route planning optimization. All restricted airspace between 15,000 and 40,000 feet are considered given the cruise altitude of approximately 30,000 feet. Meteorological hazards including thunderstorm activity, turbulence zones, and icing potential are incorporated into the path planning algorithm.

In Figure 6a, the pilot specifies a transcontinental flight from New York to Los Angeles. The start and end points are accurately recognized by the system. Given that long-range flights necessitate multiple refueling stops, NELV incorporates a flight option feature enabling pilot selection among optimization strategies. By default, the system employs a *balanced* configuration, which generates routes that optimize

the trade-off between total route distance and cumulative cost. Figure 6b presents all feasible route alternatives. The pilot retains the capability to modify flight options dynamically. In this instance, the pilot selects the *cheapest* option, prompting the system to reconfigure the flight strategy and select the corresponding cost-optimized route. The resulting path derived from this route selection is illustrated in Figure 6c. Notably, given the cruise altitude of approximately 30,000 feet for long-range operations, terrain-based risks are rendered negligible, and the path planning algorithm considers exclusively airspace restrictions.

V. ROADMAP AND FUTURE DIRECTIONS

Our current NELV system architecture comprises five discrete components to accommodate the evolving capabilities of general-purpose LLMs in UAV applications. While present-day LLMs demonstrate substantial potential, they require specialized integration with domain-specific knowledge including airspace regulations, aeronautical mapping data, UAV control systems, operational protocols, and swarm intelligence, among others. The modular design of NELV anticipates the progressive enhancement of LLM capabilities, enabling gradual consolidation of system components as UAV-specific language models mature. Furthermore, the development of comprehensive UAV datasets and specialized training methodologies will facilitate the creation of domain-optimized LLM capable of handling increasingly complex operational scenarios. As illustrated in Figure 7, we envision a systematic evolution toward unified LLM architectures that can seamlessly integrate multiple UAV operational functions. This roadmap outlines the progressive integration pathway, delineating the information requirements, decision-making capabilities, and technical prerequisites for LLMs at each automation level, ultimately leading to fully autonomous UAV mission planning and execution systems.

A. Level 1: LLM-as-Parser

Prerequisite: Prompt engineering.
Task Execution: Route planner (Algorithm 3), path planner (Algorithm 4), control platform (Algorithm 5), and safety pilot.
LLM's Output: Start point, end point, and POIs.
Responsibility: The NELV system currently is Level 1, which employs LLM as a parser to extract information from human pilot instructions. This represents a relatively straightforward application, since the LLM primarily performs linguistic understanding and reasoning to extract key information from natural language input, such as the start point and the end point designations. General-purpose LLMs adequately perform this function, as they require no specialized UAV knowledge but focus exclusively on specific token extraction. In more complex operational scenarios, LLMs must extract sophisticated constraints and requirements, exemplified when a human pilot requests "please ensure arrival before the airport closes," which contains the time constraint "airport closing." In such instances, the LLM need not comprehend the semantic meaning of "airport closing" but simply process

it as a predefined identifier for subsequent evaluation during route planning (as implemented in Algorithm 2 - Line 4). Consequently, while our system effectively performs basic information extraction, the extractable information remains limited to predefined keywords specified in the system prompt, such as the start point, the end point, time constraints, and similar elements, without the ability to identify unspecified keyword types.

Challenges: Employing LLM solely for information extraction presents significant limitations despite its proficiency in performing this relatively straightforward mission. Since we do not require the LLM to engage in cognitive processes or decision-making, it remains incapable of managing operations beyond predetermined keywords or navigating complex scenarios. For the LLM-as-Parser framework, we must predefine the specific information to be extracted in the system prompt (e.g., names of the start point and the end point). The LLM consequently fails to comprehend non-predefined information, such as "this is an urgent medicine delivery," and therefore does not incorporate this requirement in the route planning process. A further limitation is that LLM-as-Parser does not integrate contextual information to facilitate more sophisticated inference. For example, in relocation scenarios where certain airports experience adverse weather conditions such as snowstorms, aviation best practices dictate avoiding these locations. If LLM-as-Parser performs only information extraction, it cannot eliminate these compromised airports during the route planning phase. Concurrently, if the route planner does not take into account meteorological factors, the resulting routes are suboptimal. More critically, in these scenarios, human pilots cannot utilize natural language to modify the information extracted by LLM-as-Parser or adjust the planned route. Instead, human pilots must manually exclude these airports, thereby increasing their operational burden.

B. Level 2: LLM-as-Route-Planner

Prerequisites: Prompt engineering and expert feedback.
Task Execution: Path planner (Algorithm 4), control platform (Algorithm 5), and safety pilot.
LLM's Output: Route (i.e., a sequence of nodes).
Responsibility: Level 2 of the NELV system implements LLM-as-Route-Planner for route planning, necessitating that the LLM possesses comprehensive knowledge of the operational environment, specifically regarding nodes, including their geolocations, operational status, and other mission-critical parameters, such as fuel price. Beyond mere knowledge acquisition, LLM-as-Route-Planner must perform a sophisticated analysis of human pilot directives, demonstrate contextual reasoning capabilities, comprehend human pilot preferences, and formulate optimal planning solutions. For example, when given the instruction "I want to fly a UAV to a nearby airport for refueling, considering the most economical flight path," the LLM-as-Route-Planner must systematically evaluate proximate airports, their geospatial positions, fuel expenditure metrics, flight consumption algorithms, aerodynamic efficiency considerations related to wind vectors, and additional relevant parameters. In contrast, when presented
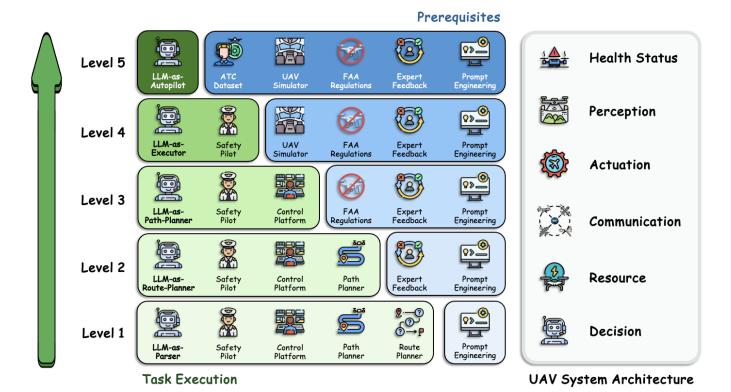
Fig. 7. Evolutionary Roadmap of Our NᴇLV System. **Green Box – Task Execution:** Components required for NᴇLV system operation, demonstrating progressive integration where higher automation levels reduce component complexity while achieving enhanced system integration, evolving from Level 1 (LLM-as-Parser) to Level 5 (LLM-as-Autopilot). **Blue Box – Prerequisites:** Infrastructure and knowledge requirements that expand with each automation level to support increasingly sophisticated LLM reasoning capabilities. **Gray Box – UAV System Architecture:** Essential UAV sensing, communication, and operational interface components that provide foundational support across all automation levels.

with the directive "proceed as quickly as possible, as I need to use it soon," the LLM-as-Route-Planner must identify the temporally optimal route. The LLM-as-Route-Planner must not only interpret the implicit objectives underlying human instructions, but also perform computational analysis to determine viable routes that satisfy these objectives.

Challenges: To fulfill these operational requirements, LLM-as-Route-Planner necessitates enhanced knowledge resources and computational capabilities for effective decision-making processes. Primarily, the system requires access to comprehensive geospatial databases containing detailed information about mission-relevant nodes, which exhibit significant quantitative variation depending on specific mission parameters and geographical contexts. For instance, according to the Yelp Open Dataset [50], Indianapolis contains 54 pharmacies and 234 supermarkets, illustrating the substantial variability in POI density across different urban environments and service categories. Additionally, LLM-as-Route-Planner must execute computational operations utilizing this information, such as calculating flight duration metrics and ranking airports according to relevant criteria. Furthermore, LLM-as-Route-Planner must demonstrate a sophisticated understanding of pilot preferences, encompassing both explicit and implicit dimensions. Explicit preferences are directly articulated within the pilot instructions, including economic efficiency considerations, temporal urgency factors, or spatial proximity requirements. Implicit preferences are derived from human experiential

judgments, which could potentially be addressed through reinforcement learning from human feedback (RLHF) [54]–[56]. LLM-as-Route-Planner might achieve these functional capabilities through specialized training protocols or fine-tuning methodologies, or potentially through integration with auxiliary APIs. Regardless of the implementation methodology, this operational level necessitates that the LLM process substantially expanded information volumes and leverage this information to execute more sophisticated and autonomous decision-making processes.

C. Level 3: LLM-as-Path-Planner

Prerequisites: Prompt engineering, expert feedback, airspace information.

Task Execution: Control platform (Algorithm 5), and safety pilot.

LLM's Output: Path (i.e., a sequence of waypoints).

Responsibility: Level 3 of the NᴇLV system implements LLM-as-Path-Planner for more detailed waypoint determination, which requires significantly more information compared to Level 2 LLM-as-Route-Planner, as it must provide waypoints beyond the nodes specified in the route. While LLM-as-Route-Planner may be considered a common-sense node selection process, waypoint determination within a path demands more extensive aviation domain knowledge, including considerations of densely populated areas, different airspace classifications, air traffic density, and other factors

to facilitate rational path planning. Similar to LLM-as-Route-Planner, LLM-as-Path-Planner must comprehend human pilot preferences, not only regarding node selection but also path features. These preferences might include specific flight times, flight paths, and other operational parameters. Another distinction from route planning is that the waypoints in the UAV mission paths are three-dimensional, encompassing latitude, longitude, and altitude. LLM-as-Path-Planner must determine not only the geographical position of the waypoints but also consider flight altitude, adhering to FAA regulations while accommodating human pilot preferences.

Challenges: Even if LLM capabilities can achieve reasonable route planning at Level 2, a substantial research gap exists in the transition to Level 3 path planning. The distinctive challenges of path planning involve understanding and complying with FAA regulations and planning appropriate flight altitudes. Consequently, LLMs require profound comprehension of FAA regulations, particularly considering that multiple factors invoke different rules, including UAV type, dimensions, weight, flight timing (e.g., nighttime operations), flight velocity, and numerous other regulatory provisions. Typically, the current NELV system requires human pilots to manually specify the UAV type (i.e., pre-configure certain UAV types and missions) to ensure compliance with FAA regulations. When directly utilizing LLM-as-Path-Planner, it must consider different FAA regulations based on human pilot language input or system background information. An additional challenge involves UAV flight altitude design. Although take-off, landing, and mission execution traffic patterns are designed within the control platform, consideration must also be given to path altitude during transit. Flight altitude planning is crucial as it affects fuel efficiency, airspace management, weather considerations, overall flight safety, and other factors.

D. Level 4: LLM-as-Executor

Prerequisite: Prompt engineering, expert feedback, airspace information, and UAV simulator.
Task Execution: Safety pilot.
LLM's Output: Trajectory (i.e., a sequence of waypoints including take-off, landing, and mission-specific flight patterns).
Responsibility: Level 4 of the NELV system further encompasses the functionality of the cloud platform, directly employing LLM-as-Executor to plan executable trajectories and *remotely control* UAV flight operations. As mentioned previously, the control platform extends beyond basic path planning to consider flight patterns, which is particularly crucial for fixed-wing UAVs that require specific patterns for take-off, landing, and mission execution. Generally, fixed-wing UAVs adhere to standard traffic patterns (i.e., left-hand patterns) for take-off and landing procedures. Consequently, LLM-as-Executor must design different traffic patterns based on airport specifications. Beyond take-off and landing considerations, LLM-as-Executor also needs to implement different designs for various mission types. For example, in a mission involving wild yak population observation, multi-rotor UAVs can hover stationary, allowing LLM-as-Executor to simply design a three-dimensional coordinate position. In contrast,

fixed-wing UAVs, which are incapable of hovering, require circular traffic patterns to facilitate continuous observation of yak populations at the same location. LLM-as-Executor necessitates a comprehensive understanding of different UAVs' traffic characteristics to effectively design take-off, landing, and mission-specific traffic patterns.

Challenges: The primary challenge involves extensive knowledge of UAVs, particularly regarding different traffic patterns for various UAV types. Similar to path planning, this is strongly correlated with UAV types and sizes. Comparatively, the challenges for LLM-as-Executor focus more on UAV-inherent limitations; for example, it cannot command vertical take-off and landing for fixed-wing UAVs, as this would result in insufficient lift and subsequent crash. Furthermore, standard take-off and landing patterns vary significantly for UAVs of different sizes and weights. Furthermore, mission-related traffic patterns constitute another key challenge, with substantial variations between different mission types. Using fixed-wing UAVs as examples, different mission-specific traffic patterns might include circular flight (e.g., search and rescue), airdrop operations (e.g., medication delivery), grid coverage (e.g., agricultural spraying), and numerous others. In practice, mission-related traffic patterns present greater difficulty compared to take-off and landing procedures, as airport information is finite and predetermined, therefore we can always preemptively collect comprehensive data on global airports for LLM-as-Executor training. However, most missions remain unpredictable and uncontrollable, such as search and rescue operations following natural disasters. LLM-as-Executor requires an exceptionally comprehensive understanding of potential missions and scenarios to conduct rational planning.

E. Level 5: LLM-as-Autopilot

Prerequisite: Prompt engineering, expert feedback, airspace information, UAV simulator, and ATC datasets.
Task Execution: LLM-as-Autopilot only.
LLM's Output: UAV actions and interpretations (to enable human pilot intervention when necessary).
Responsibility: The final level of the NELV system is LLM-as-Autopilot, representing a comprehensive integration where the LLM autonomously controls the UAV to complete missions upon receiving human pilot instructions without requiring additional human intervention. Importantly, this does not imply that the LLM directly manipulates the UAV's control surfaces. Rather, it autonomously utilizes available resources and integrated APIs for operation, including map service APIs, air traffic APIs, control APIs, and other functional interfaces. The emphasis here is on autonomous decision-making based on current situational assessment, comparable to the capabilities of a qualified human pilot making decisions based on available information. An enhanced functionality would be maintaining communication capabilities with human pilots. For example, when UAV operational conditions become complex, the system should report to the supervising human pilot and request assistance or control transfer. In addition, it must facilitate information sharing or access permissions and

communicate planned actions with other LLM-as-Autopilots. For example, in disaster search and rescue operations, multiple UAVs conducting collaborative search missions can engage in real-time sharing of search area information and coordinate subsequent search zone allocations to optimize search efficiency.

Challenges: The fundamental challenge of LLM-as-Autopilot involves LLM simulation of human pilot decision-making behaviors, specifically requiring a comprehensive understanding of all UAV systems, communication protocols with supervising human pilots, and information exchange mechanisms with other LLM-as-Autopilots. Beyond basic control functionalities, a more critical challenge emerges when the UAV encounters unpredictable contingencies, necessitating reliable response protocols. For example, when a UAV experiences bird strike incidents, the LLM-as-Autopilot must first stabilize the aircraft, subsequently assess all systems' operational status, request human pilot intervention if warranted, and finally alert other UAVs regarding the avian threat to prevent additional incidents. The LLM requires comprehensive knowledge of all UAV systems to perform effective system diagnostics, operational assessment, and implement appropriate emergency protocols, while simultaneously demanding advanced UAV-specific logical reasoning capabilities. In summary, the key challenge lies in effectively achieving human pilot-level decision-making capabilities in various operational scenarios.

VI. CONCLUSION

This paper presented the Next-Generation LLM for UAV (NELV) system, which maps natural-language tasking to short-, medium-, and long-range flight execution. NELV implements an end-to-end pipeline, from language understanding through route and path planning to control integration and real-world deployment. Across three representative case studies (short-range multi-UAV patrol, medium-range multi-POI delivery, and long-range multi-hop relocation), we demonstrated its feasibility in operational environments. The system reduces pilot workload while respecting user-defined constraints, applicable airspace regulations, and executable waypoints.

Looking ahead, we outlined a five-level roadmap for LLM-powered UAV autonomy, progressing from the current LLM-as-Parser stage to the envisioned LM-as-Autopilot with pilot-level decision making. This trajectory emphasizes the gradual integration of reasoning, planning, and control as models mature and as domain-specific datasets, regulatory codices, and communication protocols are incorporated. Key challenges remain, including the scarcity of UAV-specific datasets, the difficulty of embedding regulatory and safety knowledge into models, and the need for robust benchmarks and simulators for safety-critical evaluation. In conclusion, NELV serves both as a practical demonstration and as a forward-looking framework for LLM-enabled aerial systems, laying the foundation for UAVs that are safer, more adaptable, and more accessible to human operators.

REFERENCES

[1] D. Mahmud, H. Hajmohamed, S. Almentheri, S. Alqaydi, L. Aldhaheri, R. A. Khalil, and N. Saeed, "Integrating llms with its: Recent advances, potentials, challenges, and future directions," *IEEE Transactions on Intelligent Transportation Systems*, 2025.

[2] L. Yuan, D.-J. Han, S. Wang, and C. G. Brinton, "Local-cloud inference offloading for llms in multi-modal, multi-task, multi-dialogue settings," *arXiv preprint arXiv:2502.11007*, 2025.

[3] W. Fang, D.-J. Han, L. Yuan, and C. G. Brinton, "Collaborative device-cloud llm inference through reinforcement learning," *arXiv preprint arXiv:2509.24050*, 2025.

[4] L. Yuan, D.-J. Han, C. G. Brinton, and S. Brunswicker, "Llmap: Llm-assisted multi-objective route planning with user preferences," *arXiv preprint arXiv:2509.12273*, 2025.

[5] S. Javaid, H. Fahim, B. He, and N. Saeed, "Large language models for uavs: Current state and pathways to the future," *IEEE Open Journal of Vehicular Technology*, 2024.

[6] Y. Tian, F. Lin, Y. Li, T. Zhang, Q. Zhang, X. Fu, J. Huang, X. Dai, Y. Wang, C. Tian *et al.*, "Uavs meet llms: Overviews and perspectives toward agentic low-altitude mobility," *arXiv preprint arXiv:2501.02341*, 2025.

[7] G. Liu, N. Van Huynh, H. Du, D. T. Hoang, D. Niyato, K. Zhu, J. Kang, Z. Xiong, A. Jamalipour, and D. I. Kim, "Generative ai for unmanned vehicle swarms: Challenges, applications and opportunities," *arXiv preprint arXiv:2402.18062*, 2024.

[8] G. Sun, W. Xie, D. Niyato, H. Du, J. Kang, J. Wu, S. Sun, and P. Zhang, "Generative ai for advanced uav networking," *IEEE Network*, 2024.

[9] Z. Kaleem, F. A. Orakzai, W. Ishaq, K. Latif, J. Zhao, and A. Jamalipour, "Emerging trends in uavs: From placement, semantic communications to generative ai for mission-critical networks," *IEEE Transactions on Consumer Electronics*, 2024.

[10] F. Yao, Y. Yue, Y. Liu, X. Sun, and K. Fu, "Aeroverse: Uav-agent benchmark suite for simulating, pre-training, finetuning, and evaluating aerospace embodied world models," *arXiv preprint arXiv:2408.15511*, 2024.

[11] G. Chen, X. Yu, N. Ling, and L. Zhong, "Typefly: Flying drones with large language model," *arXiv preprint arXiv:2312.14950*, 2023.

[12] S. Liu, H. Zhang, Y. Qi, P. Wang, Y. Zhang, and Q. Wu, "Aerialvln: Vision-and-language navigation for uavs," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 15 384–15 394.

[13] R. Sinha, A. Elhafsi, C. Agia, M. Foutter, E. Schmerling, and M. Pavone, "Real-time anomaly detection and reactive planning with large language models," *arXiv preprint arXiv:2407.08735*, 2024.

[14] J. Zhong, M. Li, Y. Chen, Z. Wei, F. Yang, and H. Shen, "A safer vision-based autonomous planning system for quadrotor uavs with dynamic obstacle trajectory prediction and its application with llms," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2024, pp. 920–929.

[15] J. Cui, G. Liu, H. Wang, Y. Yu, and J. Yang, "Tpml: Task planning for multi-uav system with large language models," in *2024 IEEE 18th International Conference on Control & Automation (ICCA)*. IEEE, 2024, pp. 886–891.

[16] P. Pueyo, E. Montijano, A. C. Murillo, and M. Schwager, "Clipswarm: Generating drone shows from text prompts with vision-language models," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2024, pp. 11 917–11 923.

[17] H. Shakhatreh, A. H. Sawalmeh, A. Al-Fuqaha, Z. Dou, E. Almaita, I. Khalil, N. S. Othman, A. Khreishah, and M. Guizani, "Unmanned aerial vehicles (uavs): A survey on civil applications and key research challenges," *Ieee Access*, vol. 7, pp. 48 572–48 634, 2019.

[18] S. A. H. Mohsan, M. A. Khan, F. Noor, I. Ullah, and M. H. Alsharif, "Towards the unmanned aerial vehicles (uavs): A comprehensive review," *Drones*, vol. 6, no. 6, p. 147, 2022.

[19] FAA, "Drones by the numbers," https://www.faa.gov/node/54496, 2025.

[20] H. Kurunathan, H. Huang, K. Li, W. Ni, and E. Hossain, "Machine learning-aided operations and communications of unmanned aerial vehicles: A contemporary survey," *IEEE Communications Surveys & Tutorials*, vol. 26, no. 1, pp. 496–533, 2023.

[21] Z. Ning, H. Hu, X. Wang, L. Guo, S. Guo, G. Wang, and X. Gao, "Mobile edge computing and machine learning in the internet of unmanned aerial vehicles: a survey," *ACM Computing Surveys*, vol. 56, no. 1, pp. 1–31, 2023.

[22] D. Guo, Z. Zhang, B. Yang, J. Zhang, H. Yang, and Y. Lin, "Integrating spoken instructions into flight trajectory prediction to optimize automa-

tion in air traffic control," *Nature Communications*, vol. 15, no. 1, p. 9662, 2024.

[23] Y. Han, H. Liu, Y. Wang, and C. Liu, "A comprehensive review for typical applications based upon unmanned aerial vehicle platform," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 15, pp. 9654–9666, 2022.

[24] Y. Gao, Z. Wang, L. Jing, D. Wang, X. Li, and B. Zhao, "Aerial vision-and-language navigation via semantic-topo-metric representation guided llm reasoning," *arXiv preprint arXiv:2410.08500*, 2024.

[25] M. Chu, Z. Zheng, W. Ji, T. Wang, and T.-S. Chua, "Towards natural language-guided drones: Geotext-1652 benchmark with spatial relation matching," in *European Conference on Computer Vision*. Springer, 2024, pp. 213–231.

[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] M. Jones, S. Djahel, and K. Welsh, "Path-planning for unmanned aerial vehicles with environment complexity considerations: A survey," *ACM Computing Surveys*, vol. 55, no. 11, pp. 1–39, 2023.

[28] A. Tagliabue, K. Kondo, T. Zhao, M. Peterson, C. T. Tewari, and J. P. How, "Real: Resilience and adaptation using large language models on autonomous aerial robots," in *2024 IEEE 63rd Conference on Decision and Control (CDC)*. IEEE, 2024, pp. 1539–1546.

[29] H. Tang, Y. Zhang, V. Mohmoodian, and H. Charkhgard, "Automated flight planning of high-density urban air mobility," *Transportation Research Part C: Emerging Technologies*, vol. 131, p. 103324, 2021.

[30] W. Sribunma, L.-Y. Lin, K. A. Pant, C. Deng, J. M. Goppert, S. Brunswicker, and J. Roberts, "Mixed-reality testbed for autonomous fixed-wing uavs human-autonomy teaming," in *2024 International Conference on Assured Autonomy (ICAA)*. IEEE, 2024, pp. 94–97.

[31] V. Roberge, M. Tarbouchi, and G. Labonté, "Fast genetic algorithm path planner for fixed-wing military uav using gpu," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 54, no. 5, pp. 2105–2117, 2018.

[32] H. Wang, J. Wang, G. Ding, J. Chen, F. Gao, and Z. Han, "Completion time minimization with path planning for fixed-wing uav communications," *IEEE Transactions on Wireless Communications*, vol. 18, no. 7, pp. 3485–3499, 2019.

[33] F. Song, H. Xing, X. Wang, S. Luo, P. Dai, Z. Xiao, and B. Zhao, "Evolutionary multi-objective reinforcement learning based trajectory control and task offloading in uav-assisted mobile edge computing," *IEEE Transactions on Mobile Computing*, vol. 22, no. 12, pp. 7387–7405, 2022.

[34] H. Gong, B. Huang, and B. Jia, "Energy-efficient 3-d uav ground node accessing using the minimum number of uavs," *IEEE Transactions on Mobile Computing*, 2024.

[35] Y. Wang, J. Zhu, H. Huang, and F. Xiao, "Bi-objective ant colony optimization for trajectory planning and task offloading in uav-assisted mec systems," *IEEE Transactions on Mobile Computing*, 2024.

[36] M. Zhao, R. Zhang, Z. He, and K. Li, "Joint optimization of trajectory, offloading, caching, and migration for uav-assisted mec," *IEEE Transactions on Mobile Computing*, 2024.

[37] G. Sun, Y. Wang, Z. Sun, Q. Wu, J. Kang, D. Niyato, and V. C. Leung, "Multi-objective optimization for multi-uav-assisted mobile edge computing," *IEEE Transactions on Mobile Computing*, 2024.

[38] C. Deng, W. Sribunma, S. Brunswicker, J. M. Goppert, and I. Hwang, "3d path planning with weather forecasts, ground risks, and airspace information for uav mid-mile delivery," in *AIAA SCITECH 2025 Forum*, 2025, p. 1806.

[39] F. A. Administration, *Airplane flying handbook (FAA-H-8083-3A)*. Skyhorse Publishing Inc., 2011.

[40] Mission Control, "Windracers Mission Control," https://windracers.com/ultra/#software, 2025.

[41] OpenAI, "Hello gpt-4o," https://openai.com/index/hello-gpt-4o/, 2024.

[42] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95 - International Conference on Neural Networks*, vol. 4, 1995, pp. 1942–1948 vol.4.

[43] AirNav, LLC, "AirNav: Airport, FBO & Fuel Prices Information," https://www.airnav.com/, 2025.

[44] Riverbank Computing, "PyQt5," https://www.riverbankcomputing.com/static/Docs/PyQt5/, 2025.

[45] A. Abouelenin, A. Ashfaq, A. Atkinson, H. Awadalla, N. Bach, J. Bao, A. Benhaim, M. Cai, V. Chaudhary, C. Chen *et al.*, "Phi-4-mini technical report: Compact yet powerful multimodal language models via mixture-of-loras," *arXiv preprint arXiv:2503.01743*, 2025.

[46] Windracers, "Windracers ULTRA," https://windracers.com/ultra/, 2025.

[47] PX4 Development Team, "PX4 Autopilot," https://px4.io/, 2025.

[48] ArduPilot, "Mission Planner Overview," https://ardupilot.org/planner/, 2024.

[49] OpenStreetMap contributors, "Planet dump retrieved from https://planet.osm.org ," https://www.openstreetmap.org, 2017.

[50] Yelp Inc., "Yelp open dataset," https://business.yelp.com/data/resources/open-dataset/, accessed: 2025-09-07.

[51] OpenAIP, "OpenAIP," https://www.openaip.net/.

[52] Herbie, "Herbie: Retrieve NWP Model Data," https://herbie.readthedocs.io/en/stable/, 2025.

[53] V. Furnon and L. Perron, "Or-tools routing library," Google. [Online]. Available: https://developers.google.com/optimization/routing/

[54] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray *et al.*, "Training language models to follow instructions with human feedback," *Advances in neural information processing systems*, vol. 35, pp. 27 730–27 744, 2022.

[55] Y. Sun, N. Salami Pargoo, P. Jin, and J. Ortiz, "Optimizing autonomous driving for safety: A human-centric approach with llm-enhanced rlhf," in *Companion of the 2024 on ACM International Joint Conference on Pervasive and Ubiquitous Computing*, 2024, pp. 76–80.

[56] Z. Huang, Z. Sheng, and S. Chen, "Trustworthy human-ai collaboration: Reinforcement learning with human feedback and physics knowledge for safe autonomous driving," *arXiv preprint arXiv:2409.00858*, 2024.