

SLM, LLM or Agentic AI? Toward Intelligent UAV-Enabled WPT Systems in Low-Altitude Economy Networks

Feibo Jiang, *Senior Member, IEEE*, Li Dong, Lei Mao, Kezhi Wang, *Senior Member, IEEE*, Xianbin Wang, *Fellow, IEEE*, Abbas Jamalipour, *Fellow, IEEE*.

Abstract—Unmanned Aerial Vehicles (UAVs) have become a key enabling technology for low-altitude economic networks, yet achieving efficient and adaptive optimization under resource-constrained and dynamic environments remains challenging. This paper investigates language models for UAV-enabled Wireless Power Transfer (WPT) systems. First, a lightweight SLM-based solution is developed using a pre-trained BERT backbone, enhanced UAV embeddings and contextual features, a geometry-aware path decoder, and ensemble inference to achieve low complexity, low latency, and high energy efficiency. Second, an Agentic AI-based framework is designed to exploit the reasoning and interactive capabilities of LLMs. It integrates four collaborative agents—Initializer, Actor, Critic, and Reflector—to form a closed loop of generation, optimization, evaluation, and reflection for iterative UAV path and energy optimization. Finally, simulations compare the SLM-, LLM-, and Agentic AI-based approaches.

Index Terms—Small Language Model, Large Language Model, Agentic AI, Unmanned Aerial Vehicle, Wireless Power Transfer, Path Planning

I. INTRODUCTION

A. Background

In recent years, the Low-Altitude Economy Network (LAENet) has emerged as a new frontier of future information infrastructure and economic growth. It is gradually evolving into a multi-layered network that integrates transportation, energy supply, air-ground collaboration, and intelligent services. Within LAENet, Unmanned Aerial Vehicle (UAV) has become

a core node supporting network operation and value realization due to its high mobility, rapid deployment, and wide-area coverage capabilities. However, UAV systems also face multi-dimensional challenges related to limited computing, communication, and energy resources, which place higher demands on their ability to operate over extended periods.

Energy consumption is a critical bottleneck that constrains both the flight endurance and service capabilities of UAVs. Due to payload and size limitations, the onboard battery capacity of UAVs is inherently restricted, making it difficult to sustain long-duration missions and large-scale networked applications [1]. To address this issue, Wireless Power Transfer (WPT) technology has been integrated into UAV systems [2]. WPT enables UAVs to receive energy replenishment during flight and, at the same time, function as energy nodes that provide wireless power to ground terminals or other devices. This approach effectively overcomes the limitations of traditional battery-powered modes. By extending individual UAV flight time and enabling sustained operation of multi-UAV and terminal networks, WPT significantly improves overall task completion rates and resource utilization efficiency.

B. Motivation

Path planning plays a crucial role in UAV-enabled WPT system architectures. During mission execution, UAVs must simultaneously consider multiple factors, including flight energy consumption, energy replenishment locations, communication coverage, and mission delay. These interrelated requirements make path planning a highly complex optimization problem. On one hand, dynamic environmental changes, uncertain energy transfer efficiency, and diverse task demands introduce significant uncertainty and high-dimensional coupling into the UAV path optimization process. On the other hand, traditional methods, which mainly rely on heuristic algorithms [3] and convex optimization algorithms [4], can improve path selection and energy efficiency to some extent but often suffer from slow convergence, high computational complexity, and limited adaptability to dynamic environments. Achieving energy-efficient, intelligent, and robust path planning remains a key research focus and a major challenge in this field.

In recent years, the rapid progress of Large Language Models (LLMs) has introduced promising opportunities for enhancing the optimization of UAV-enabled WPT systems, owing to their powerful capabilities in complex reasoning,

This work was supported in part by the National Natural Science Foundation of China under Grants 62572184, and 41604117; in part by the Natural Science Foundation of Hunan Province under Grants 2024JJ5270 and 2025JJ50365; in part by the Changsha Natural Science Foundation under Grants kq2402098 and kq2402162 (Corresponding author: Li Dong.)

Feibo Jiang (jiangfb@hunnu.edu.cn) is with the Hunan Provincial Key Laboratory of Intelligent Computing and Language Information Processing and the Yuelushan Digital Intelligence Laboratory (Artificial Intelligence and International Communication), Hunan Normal University, Changsha 410081, China.

Li Dong (Dlj2017@hunnu.edu.cn) is with the School of Computer Science, Hunan University of Technology and Business, Changsha 410205, China, and also with the Xiangjiang Laboratory, Changsha 410205, China.

Lei Mao (202520294367@hunnu.edu.cn) is with School of Information Science and Engineering, Hunan Normal University, Changsha 410081, China.

Kezhi Wang (Kezhi.Wang@brunel.ac.uk) is with the Department of Computer Science, Brunel University London, London UB8 3PH, UK.

Xianbin Wang (xianbin.wang@uwo.ca) is with the Department of Electrical and Computer Engineering, Western University, London N6A 5B9, Canada.

Abbas Jamalipour (a.jamalipour@ieee.org) is with the School of Electrical and Computer Engineering, University of Sydney, Australia, and with the Graduate School of Information Sciences, Tohoku University, Japan.

strategic planning, and cross-modal understanding [5]. By enabling natural language interaction, knowledge transfer, and adaptive optimization, LLMs can intelligently interpret dynamic task environments and support efficient decision-making. This makes them a promising tool for enhancing multi-UAV collaboration and energy-efficient path planning [6]. However, the significant training and inference costs of LLMs pose critical challenges. Effectively integrating LLM-based intelligent reasoning capabilities while maintaining system energy efficiency and accommodating resource constraints remains a key issue for future research.

C. Contributions

Against this backdrop, this paper proposes and systematically investigates three intelligent solution frameworks for UAV-enabled WPT systems in LAENets, aiming to strike a balance between model performance and resource consumption. The contributions of this study are summarized as follows.

1) *SLM-Based Lightweight Optimization*: To address the resource constraints of energy-efficient UAV-enabled WPT systems, we propose a lightweight SLM-based solution. Built on a pre-trained BERT backbone, it integrates enhanced UAV embeddings, contextual features, a geometry-aware path decoder, ensemble inference, and reinforcement learning-based optimization. These designs improve path-geometry modeling and optimization performance. The resulting SLM-based approach achieves low computational complexity, high energy efficiency, and low inference latency, making it suitable for real-time and resource-constrained UAV operations in LAENets.

2) *LLM- and Agentic AI-Based Reasoning*: To handle the uncertainty and dynamic coupling between path planning and energy consumption, we further explore the reasoning and interactive capabilities of LLMs through an Agentic AI framework. This framework includes four cooperative agents, namely Initializer, Actor, Critic, and Reflector, forming a closed loop of solution generation, optimization, evaluation, and feedback refinement. In addition, the Critic uses the Model Context Protocol (MCP) to invoke external tools for constraint verification and local path search, thereby improving generalization and adaptability in dynamic environments.

3) *Cross-Paradigm Evaluation and Complementarity*: Through theoretical analysis and simulations, we compare the SLM-, LLM-, and Agentic AI-based schemes in terms of energy efficiency, path quality, and computational complexity. Results show that SLM has clear advantages in resource-constrained scenarios, whereas Agentic AI provides stronger decision-making intelligence and robustness in dynamic environments. Further analysis reveals their complementary roles and applicability boundaries in LAENets, offering insights into the integration of lightweight modeling and intelligent reasoning for future wireless systems.

D. Organization

The remainder of this paper is organized as follows. Section II reviews the related work. Section III presents the system model and problem formulation. Section IV introduces the

SLM-based solution framework, while Section V elaborates on the LLM and Agentic AI-based solution framework. Section VI reports and analyzes the experimental results. Finally, Section VII concludes the paper.

II. RELATED WORK

A. Energy-Efficient UAV-Enabled WPT Systems

Yang et al. [7] proposed a Dueling Double Deep Q-Network (Dueling DDQN)-based UAV-IoT system that jointly optimizes trajectory, power allocation, and scheduling in a Simultaneous Wireless Information and Power Transfer (SWIPT) scenario. Dong et al. [8] proposed an Attention-based UAV Trajectory Optimization (AUTO) framework for WPT-assisted IoT systems, achieving multi-UAV trajectory and deployment optimization. By integrating the Attention-based Trajectory Optimization Model (ATOM), the framework enhances both energy efficiency and stability, effectively addressing trajectory optimization challenges in large-scale IoT networks. Muy et al. [9] proposed a grid-world Deep Reinforcement Learning (DRL)-based UAV-WPT system that jointly optimizes trajectory, beamforming, and power allocation. Kim et al. [10] proposed a multi-UAV WPT communication network that jointly optimizes trajectory, scheduling, and power allocation. The method enhances minimum uplink throughput under energy neutrality and mobility constraints while mitigating interference in multi-UAV scenarios.

B. UAV System Optimization Based on DRL

Ning et al. [11] proposed a DRL-based UAV trajectory optimization algorithm, in which each service provider's UAV makes flight decisions solely based on local observations. The objective is to simultaneously minimize the short-term computational cost of ground users and the long-term computational cost of UAVs under conditions of incomplete information. Su et al. [12] proposed a dual-UAV secure communication system based on the Deep Q-Networks (DQN). This system achieves joint optimization of trajectory, power, and user scheduling, thereby improving the secrecy rate and convergence speed while effectively addressing physical-layer security and anti-eavesdropping challenges in UAV communications. Li et al. [13] proposed a Soft Actor-Critic for Covert Communication and Charging (SAC-CC) system, which performs joint optimization of UAV trajectory and bandwidth allocation.

C. UAV System Optimization Based on LLM

Samma et al. [14] proposed an LLM-based UAV visual path planning system that enables autonomous navigation in indoor environments where GPS signals are unavailable through visual perception. Jiang et al. [15] proposed an Agentic AI-based framework for multi-UAV trajectory optimization in LAENet. Integrating Agentic RAG, Transformer and Mamba, and Trajectory-Group Relative Policy Optimization (T-GRPO), the framework enhances energy efficiency, convergence, and generalization in dynamic environments. Yin et al. [16] proposed a Multi-Modal Large Language Model (M2LLM)-driven DRL framework for trajectory design and beamforming optimization in UAV networks. This approach improves

spectrum efficiency, energy efficiency, and convergence speed, effectively addressing the joint optimization challenges of UAVs in complex environments.

Therefore, although existing studies have achieved significant progress in energy-efficient optimization and task scheduling, current approaches still exhibit notable limitations in complex and dynamic scenarios. Specifically, traditional optimization struggles with high-dimensional, multi-objective search spaces; DRL still faces challenges in generalization and stability; and existing LLM-based studies have yet to establish deployable multi-agent reasoning frameworks. These limitations reveal critical gaps in the current literature and further underscore the necessity of developing a unified SLM-LLM-Agent AI framework. Consequently, a systematic comparison between SLM-based methods and LLM-driven Agent AI paradigms is of great significance for clarifying their complementary strengths, guiding model selection and optimization under different scenarios, and advancing the intelligent evolution of UAV-enabled WPT systems.

Table I: Comparison with previous works

Work	UAV	WPT	DRL/RL	LLM / Agent AI
[7]–[10]	✓	✓	✓	
[11]–[13]	✓		✓	
[14]–[16]	✓		✓	✓
Our Work	✓	✓	✓	✓

III. SYSTEM MODEL AND PROBLEM FORMULATION

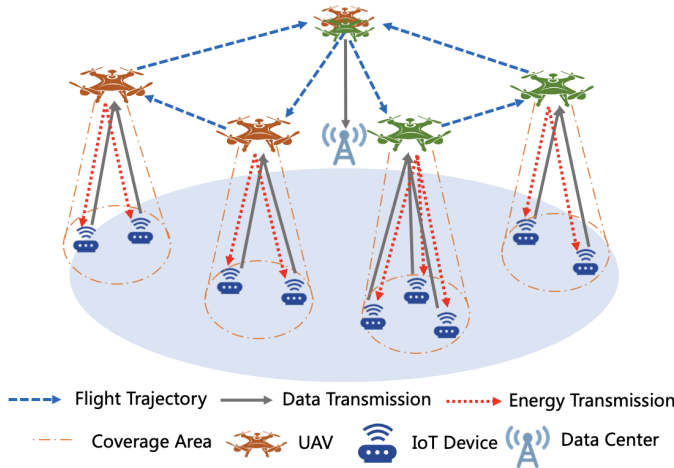


Figure 1: UAV-enabled WPT System with Two UAVs and Their Flight Paths for IoT Data Collection.

As illustrated in Fig. 1, the UAV-enabled WPT system consists of N IoTs, a data center, and m UAVs with half-duplex access points, which can transmit power to the IoTs and collect data from IoTs by Time Division Duplexing (TDD) mode. The N IoTs are denoted as a set of $\mathcal{N} = \{1, 2, \dots, N\}$. We assume that the GPS is equipped in each IoT and the position (x_i, y_i) of the i -th IoT is fixed and known. The i -th IoT has D_i data to be collected. The set of m UAVs is

denoted as $\mathcal{M} = \{1, 2, \dots, m\}$, and each UAV has limited data storage capacity $C_{\max}$ and energy capacity $E_{\max}$. The flight height of the UAV is set to H_z meters. The data of each IoT could be collected by one UAV at one time, so the association a_{ij} at the t -th time step can be expressed by:

$$a_{ij}[t] = \{0, 1\}, \forall i \in \mathcal{N}, \forall j \in \mathcal{M} \quad (1)$$

where $a_{ij}[t] = 1$ means the j -th UAV is collecting the data from the i -th IoT at the t -th time step, and $a_{ij}[t] = 0$ otherwise. Assuming each IoT can only be serviced once, then:

$$\sum_{j=1}^m \sum_{t=1}^{s_j} a_{ij}[t] = 1. \quad (2)$$

where s_j is the total number of time steps.

A. Communication Model

Considering obstacles in the system, the simplified Line-of-Sight (LoS)-dominant channel model can be inaccurate in practice [17]. In order to characterize the UAV-to-IoTD channel state more accurately, we adopt the probabilistic LoS channel model [18], where LoS probability is a function of the elevation angle between UAV and IoTs. A generalized logistic function is introduced to approximate the LoS probability as follows [17]:

$$p_{ij}^L[t] = B_3 + \frac{B_4}{1 + e^{-(B_1 + B_2 \theta_{ij}[t])}} \quad (3)$$

where $\theta_{ij}[t]$ denotes the elevation angle between the j -th UAV and the i -th IoT in time step t . B_1, B_2, B_3, B_4 are environment-related constants. Then, the corresponding non-LoS (NLoS) probability is obtained as $p_{ij}^N[t] = 1 - p_{ij}^L[t]$. Next, the large-scale channel power gain between the i -th IoT and the j -th UAV in time step t , including both the path loss and shadowing, can be approximated as

$$h_{ij}[t] = \begin{cases} \beta_0 (d_{ij})^{-\alpha_L}, & \text{LoS} \\ \mu \beta_0 (d_{ij})^{-\alpha_N}, & \text{NLoS} \end{cases} \quad (4)$$

where β_0 is the average channel power at a reference distance of 1 meter in the LoS state. μ is the signal attenuation factor due to the NLoS propagation. α_L and α_N are the average path loss exponents for LoS and NLoS states in time step t , respectively. d_{ij} is the direct distance between the i -th IoT and the j -th UAV.

B. Data Collection Model

In the proposed system, UAVs can either charge multiple IoTs simultaneously or collect data from multiple IoTs simultaneously. Hence, the whole process can be divided into two stages: the energy transfer stage and the data transmission stage.

1) *Energy Transfer Stage*: Energy transfer technology is applied in the energy transfer stage, where the phased antenna array on UAVs would generate multiple beams to charge multiple IoTs at the same time [19]. By charging multiple devices simultaneously, it greatly improves the efficiency of energy transfer.

In the system, the j -th UAV can transmit the energy to the i -th IoT wirelessly via Radio Frequency (RF) signals with transmit power P^T . We assume the uplink power gain, downlink power gain, and the large-scale channel power gain are equal [20], which is described in Eq. (4). Hence, the power received at the i -th IoT from the j -th UAV in time step t is denoted as $P_{ij}^R[t]$ as follows:

$$P_{ij}^R[t] = h_{ij}[t]P^T. \quad (5)$$

According to the Eq. (4), the expected receiving power is given as

$$P_{ij}^R[t] = p_{ij}^L[t]\beta_0(d_{ij})^{-\alpha_L}P^T + (1 - p_{ij}^L[t])\mu\beta_0(d_{ij})^{-\alpha_N}P^T. \quad (6)$$

We assume the hovering point of UAVs are set as the Cluster Heads (CHs) for a group of IoTs, and IoTs belonging to one CH would be collected by the specific UAV at the top of the CH [21]. The proposed system adopts the linear energy transfer model, and the received energy of the i -th IoT from the j -th UAV in time step t can be given by

$$E_{ij}^R[t] = \eta^L P_{ij}^R[t]T_{ij}^E[t] \quad (7)$$

where η^L is the constant attenuation parameter in the energy transfer model. $T_{ij}^E[t]$ is the energy transfer time of the i -th IoT from the j -th UAV in time step t .

2) *Data Transmission Stage*: We introduce Orthogonal Frequency Division Multiplexing (OFDM) technology for the UAV to collect data from multiple IoTs within its communication range simultaneously. The uploading data rate of the i -th IoT to the j -th UAV in time step t can be given by

$$R_{ij}[t] = B \log_2 \left(1 + \frac{h_{ij}[t]E_{ij}^R[t]}{\sigma^2 T_{ij}^C[t]} \right) \quad (8)$$

where B is the bandwidth, σ^2 is Gaussian white noise power, and $T_{ij}^C[t]$ is the data collection time from i -th IoT to the j -th UAV in time step t .

To ensure that IoTs can successfully upload their data to the UAVs, one has

$$T_{ij}^C[t]R_{ij}[t] \geq D_i, \forall j \in \mathcal{M} \quad (9)$$

where D_i is the collected data from the i -th IoT.

C. Flight Model

In the proposed system, each UAV flies in a straight line from one hovering point to another. The j -th UAV takes off from the data center at a fixed location $r_j[0] = (0, 0, H_z)$, flies to each CH as a designated hovering point, and hovers above it to collect data from all IoTs within the corresponding cluster. The j -th UAV completes the data collection task according to a predetermined flight trajectory, which consists of several

hovering points, and returns to the same data center after one flying cycle. Hence, we have

$$r_j[s_j] = r_j[0], \forall j \in \mathcal{M} \quad (10)$$

where $r_j[t]$ denotes the t -th time step of the j -th UAV, $t \in \mathcal{T}_j = \{1, 2, \dots, s_j\}$, and the trajectory of the j -th UAV has s_j hovering point.

We assume that each UAV flies at a constant speed v_j . Since each UAV flies from one hovering point to another in a straight line, the flight time of the j -th UAV can be expressed by:

$$T_j^F = \sum_{t=1}^{s_j} \frac{\|r_j[t] - r_j[t-1]\|_2}{v_j}, \forall j \in \mathcal{M} \quad (11)$$

where $\|r_j[t] - r_j[t-1]\|_2$ is the Euclidean distance between hovering point $r_j[t]$ and $r_j[t-1]$.

D. Energy Consumption Model

Assuming that the flight power P_j^F of the j -th UAV is constant, the flight energy consumption of the j -th UAV can be expressed as follows.

$$E_j^F = P_j^F T_j^F \quad (12)$$

where T_j^F is the total flight time of the j -th UAV.

Similarly, assuming that the hovering power P_j^H of the j -th UAV is constant, the hovering energy consumption of the j -th UAV can be expressed as follows.

$$E_j^H = \sum_{t=1}^{T_j} (P^H T_j^E[t] + P^H T_j^C[t]) \quad (13)$$

where $T_j^E[t] = \max_{i \in \mathcal{C}_k} T_{ij}^E[t]$ is the maximum power transfer time for IoTs in the k -th cluster, and $\mathcal{C}_k$ is the k -th cluster of IoTs. $T_j^C[t] = \max_{i \in \mathcal{C}_k} T_{ij}^C[t]$ is the maximum data collection time for IoTs in the k -th cluster, and each IoT applies the OFDM channel.

When the j -th UAV hovers at the top of one CH, it transfers its power to charge IoTs first and then collects data from them. Hence, the power transfer energy of the j -th UAV is

$$E_j^T = \sum_{t=1}^{T_j} P^T T_j^E[t] \quad (14)$$

Therefore, the total energy consumption of the j -th UAV can be given as

$$E_j = E_j^F + E_j^H + E_j^T, \forall j \in \mathcal{M}. \quad (15)$$

Due to the limited data and energy capacity of UAVs, it is required that the total energy consumption of the UAV does not exceed its energy capacity, and all collected data does not exceed the storage capacity. Therefore, these inequalities need to be satisfied

$$E_j \leq E_{\max}, \forall j \in \mathcal{M}, \quad (16)$$

$$\sum_{i=1}^N \sum_{t=1}^{s_j} a_{ij}[t]D_i \leq C_{\max}, \forall j \in \mathcal{M}. \quad (17)$$

E. Problem Formulation

We aim to minimize the energy consumption of all the UAVs by jointly optimizing the number of UAVs, the trajectories of UAVs, user association, the energy transfer time, and the data collection time. The optimization problem can be mathematically formulated by

$$P0 : \min_{\mathcal{A}, m, \mathcal{R}, \mathcal{T}^E, \mathcal{T}^C} \sum_{j=1}^m E_j \quad (18)$$

$$s.t. (1), (2), (9), (10), (16), (17)$$

where $\mathcal{A} = \{a_{ij}[t], \forall i \in \mathcal{N}, j \in \mathcal{M}, t \in \mathcal{T}_j\}$ represents the association between UAVs and IoTs. m is the optimal number of UAVs for data collection. $\mathcal{R} = \{r_j[t], \forall j \in \mathcal{M}, t \in \mathcal{T}_j\}$ represents the visiting order of hovering points for UAVs. $\mathcal{T}^E = \{T_{ij}^E, \forall i \in \mathcal{N}, j \in \mathcal{M}, t \in \mathcal{T}_j\}$ represents the set of energy transfer time and $\mathcal{T}^C = \{T_{ij}^C, \forall i \in \mathcal{N}, j \in \mathcal{M}, t \in \mathcal{T}_j\}$ represents the set of data collection time. Problem $P0$ is a Mixed Integer Non-Linear Program (MINLP), where E_j^F is affected by the visiting order of hovering point, E_j^H and E_j^T are influenced by energy transfer time and data collection time.

F. Problem Decomposition

In problem $P0$, there are an overwhelming number of terms that need to be optimized; however, it can be separated into different sub-problems which can be solved one at a time. First, we assume the number of hovering points is K , and we denote $\mathcal{H} = \{H_1, \dots, H_K\}$ as the cluster headset, where H_i denotes the coordinates of the i -th hovering point. Then, we use the large-scale path-loss fuzzy c-means clustering algorithm (LS-FCM) [22] to optimize the hovering points of UAVs and the association $\mathcal{A}$, which is based on the large-scale path-loss factors. This indicates that the proposed framework adopts a hierarchical solving strategy, where the hovering points are optimized first, followed by path planning and resource scheduling. Then, we decompose the problem $P0$ into two sub-problems: energy transfer and data collection time optimization $P1$, and path optimization $P2$.

Since the energy transfer time and the data collection time of each IoT are independent of the trajectories of UAVs, Problem $P1$ can be described as follows:

$$P1 : \min_{\mathcal{T}^E, \mathcal{T}^C} \sum_{j=1}^m (E_j^H + E_j^T) \quad (19)$$

$$s.t. (9).$$

Meanwhile, Problem $P2$ can be described as follows:

$$P2 : \min_{m, \mathcal{R}} \sum_{j=1}^m E_j^F \quad (20)$$

$$s.t. (10), (16), (17).$$

Problem $P2$ can be formulated as a well-known combinatorial optimization problem called the Multiple Traveling Salesman Problem (MTSP), which is known to be NP-hard. Moreover, each UAV has its limitations, and the number of UAVs is also required to be optimized. Hence, it is difficult to find the optimal solution by known methods.

IV. SLM-BASED SOLUTION FRAMEWORK

By combining SLMs with a path decoder, rapid UAV path planning and time allocation can be achieved at the network edge with low latency, low energy consumption, and high determinism, thus improving the performance-efficiency trade-off in resource-constrained scenarios. Therefore, we propose an SLM scheme based on a lightweight BERT model [23]. Since the original BERT architecture is designed for language data, we adapt its embedding layer, path decoder, and training strategy to suit energy-efficient UAV-enabled WPT systems. The proposed SLM is illustrated in Fig. 2.

A. Enhanced UAV Embedding

1) *Positional Feature Enhancement*: In the BERT-based model, each node on the trajectory is typically represented by its three-dimensional coordinates (x_i, y_i, z_i) . While this representation encodes the absolute spatial position of each node, it lacks explicit geometric cues such as orientation and angular relationships in 3D space. Consequently, the model has to rely solely on Euclidean distances to implicitly infer the relative spatial structures among nodes, which increases the difficulty of geometric reasoning and trajectory reconstruction.

To address this limitation, we introduce directional features (θ_i, ϕ_i) into the input, representing the **azimuth** and **elevation** angles of each node with respect to the global coordinate system:

$$\theta_i = \text{atan2}(y_i, x_i) \quad (21)$$

$$\phi_i = \text{atan2} \left(z_i, \sqrt{x_i^2 + y_i^2} \right) \quad (22)$$

where θ_i denotes the azimuth angle of node i in the x - y plane, while ϕ_i represents its elevation angle relative to the horizontal plane. These angular descriptors provide the model with richer geometric context and facilitate more accurate spatial reasoning in trajectory modeling.

The enhanced node feature vector is thus obtained as:

$$f_i = [x_i, y_i, \theta_i, \phi_i] \quad (23)$$

and is then mapped through a linear embedding layer to an initial representation compatible with the BERT model:

$$\mathbf{H}_i^{(0)} = \mathbf{W}^E f_i \quad (24)$$

where $\mathbf{W}^E$ denotes a learnable embedding matrix.

This feature enhancement allows the encoder to not only capture the spatial distribution of nodes but also recognize their geometric orientation relative to the origin. Consequently, the model's understanding of path geometry is improved, enabling the self-attention mechanism to more easily learn the relative positional patterns of all hovering points.

2) *Geometric Feature Enhancement*: The MTSP inherently exhibits rotational and reflective symmetry in geometric space; that is, the optimal length of any path remains unchanged when the entire set of points undergoes planar rotation or mirroring. This property provides an effective means for data augmentation. Specifically, we employ the square symmetry group

(i.e., the dihedral group D_4) [24] to apply the following eight geometric transformations to the original set of coordinates:

$$\begin{aligned}
 g_1 : (x, y) &\mapsto (x, y), & \text{Identity (no transformation)} \\
 g_2 : (x, y) &\mapsto (-x, y), & \text{Reflection over the } y\text{-axis} \\
 g_3 : (x, y) &\mapsto (x, -y), & \text{Reflection over the } x\text{-axis} \\
 g_4 : (x, y) &\mapsto (-x, -y), & \text{Rotation by } 180^\circ \\
 g_5 : (x, y) &\mapsto (y, x), & \text{Reflection over the line } y = x \\
 g_6 : (x, y) &\mapsto (-y, x), & \text{Rotation by } 90^\circ \\
 g_7 : (x, y) &\mapsto (y, -x), & \text{Rotation by } 270^\circ \\
 g_8 : (x, y) &\mapsto (-y, -x), & \text{Reflection over the line } y = -x.
 \end{aligned} \tag{25}$$

Based on the above geometric transformations, each topological graph instance can be converted into eight equivalent instances. All instance features are then used as the input to the initial embedding layer. Theoretically, these symmetry-enhanced augmentations preserve the optimal path and underlying policy landscape, thereby maintaining policy consistency while increasing training diversity and improving generalization [25]. Through positional and geometric feature enhancement, the expressive capacity of node embeddings is significantly enriched, improving BERT's ability to capture the geometric characteristics of paths.

B. BERT Encoder

The BERT model consists of L encoder layers. In the ℓ -th encoder layer, the Multi-Head Self-Attention (MHSA) is first computed as [26]:

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{Softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}}\right) \mathbf{V} \tag{26}$$

where the query $\mathbf{Q} = \mathbf{H}^{(\ell-1)}\mathbf{W}^Q$, key $\mathbf{K} = \mathbf{H}^{(\ell-1)}\mathbf{W}^K$, and value $\mathbf{V} = \mathbf{H}^{(\ell-1)}\mathbf{W}^V$, with $\mathbf{W}^Q, \mathbf{W}^K, \mathbf{W}^V$ being learnable weight matrices. For H attention heads, the outputs are concatenated and projected as:

$$\text{MHSA}(\mathbf{H}^{(\ell-1)}) = [\text{head}_1; \dots; \text{head}_H]\mathbf{W}^O \tag{27}$$

where $\mathbf{W}^O$ is a learnable matrix, and $\text{head}_h = \text{Attention}(\mathbf{Q}_h, \mathbf{K}_h, \mathbf{V}_h)$.

Next, residual connection (Add) and layer normalization (Norm) are applied:

$$\mathbf{Z}^\ell = \text{Norm}\left(\mathbf{H}^{(\ell-1)} + \text{MHSA}(\mathbf{H}^{(\ell-1)})\right). \tag{28}$$

The output is then passed through a Feed-Forward Network (FFN), followed by another residual connection and Layer-Norm:

$$\mathbf{H}^\ell = \text{Norm}(\mathbf{Z}^\ell + \text{FFN}(\mathbf{Z}^\ell)). \tag{29}$$

By stacking L layers of such encoders, BERT can obtain bidirectional contextual feature representations, ultimately producing $\mathbf{H}^L$, where each row corresponds to the contextual representation of a node in the input sequence.

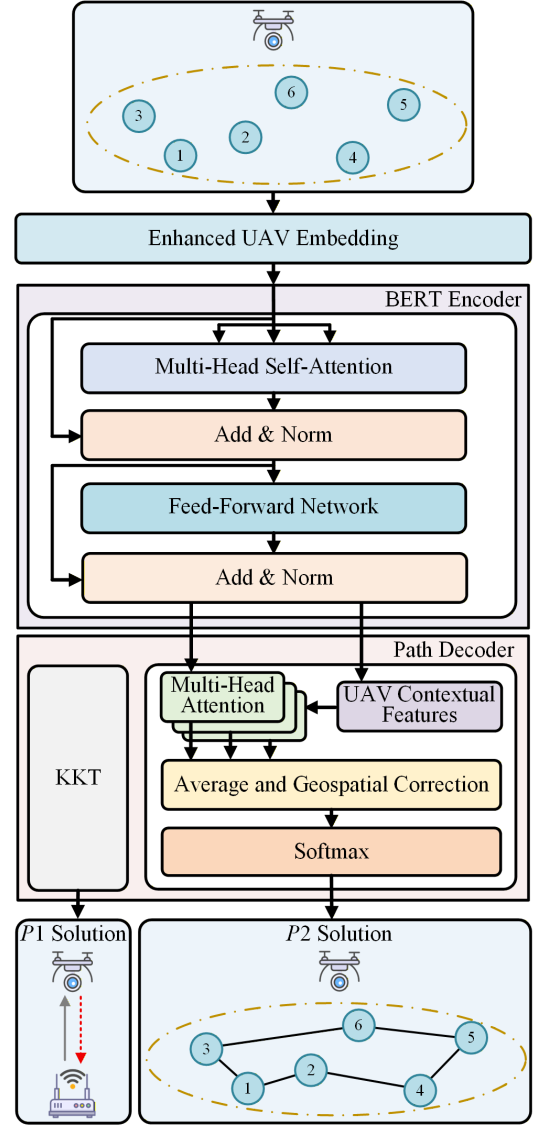


Figure 2: The SLM-Based Solution Framework.

C. Enhanced UAV Contextual Features

For high-quality path planning, it is crucial to construct a query vector $\mathbf{q}[t]$ based on the encoder output $\mathbf{H}^L$, which fully captures the current path state and guides the selection of the next node. Unlike approaches that rely solely on features derived from the “current node and global node mean” [27], we explicitly incorporate four types of feature information—starting node, previous node, global structure, and path structure—to form an enhanced UAV contextual feature representation.

1) *Starting Node Feature Vector*: $\mathbf{h}_{\pi_{[1]}}$ represents the feature vector of the starting node (typically the data center). Incorporating this ensures that the SLM model is aware of the overall starting constraint during path planning, thereby helping maintain route closure and overall feasibility when generating paths.

2) *Previous Node Feature Vector*: $\mathbf{h}_{\pi_{[t-1]}}$ represents the feature vector of the most recently visited node along the current path. It provides local positional information, serving

as the most direct spatial reference for the decoder when selecting the next node to visit.

3) *Global Feature Vector*: $\mathbf{h}_g = \frac{1}{N} \sum_{i=1}^N \mathbf{h}_i$ represents the global structure of all nodes. This pooled representation captures the overall layout, providing a macroscopic perspective for decision-making and preventing the path from relying solely on local information, which could lead to greedy choices.

4) *Path Feature Vector*: $\mathbf{h}_\pi = \sum_{s=1}^{t-1} \mathbf{h}_{\pi[s]}$ encodes the set of nodes already visited along the path, allowing the model to explicitly consider the coverage and cumulative cost of the historical trajectory. This enhances global consistency and robustness in path planning.

By aggregating the above feature vectors, the enhanced UAV contextual feature query is formulated as:

$$\mathbf{q}[t] = \frac{1}{t-1} (\mathbf{h}_\pi) + \mathbf{h}_g + \mathbf{h}_{\pi[t-1]} + \mathbf{h}_{\pi[1]}. \quad (30)$$

As a result, the query vector $\mathbf{q}[t]$ semantically incorporates global, local, and historical contextual information, significantly enhancing the decision-making capability of the subsequent path decoder. Intuitively, this construction prevents the SLM model from making decisions based solely on the current node and global mean, thereby better balancing overall path feasibility and local optimality, and effectively improving the quality and generalization of path planning.

D. Path Decoder

We design a geometry-aware path decoder to sequentially generate UAV paths. In the path decoder, the query vector $\mathbf{q}[t]$ interacts with the contextual features of candidate nodes to compute the selection probability for each node, enabling step-wise path generation.

During the decoding phase, the query vector $\mathbf{q}[t]$ and the encoding $\mathbf{h}_j$ of the j -th candidate node are first projected via separate multi-head linear projections. Scaled dot-product attention is then applied to compute multi-head interaction scores. Averaging over H attention heads yields the attention score for the j -th candidate node:

$$s_j[t] = \frac{1}{H} \sum_{h=1}^H \frac{(\mathbf{q}[t] \mathbf{W}_h^Q)(\mathbf{h}_j \mathbf{W}_h^K)^T}{\sqrt{d_k}} \quad (31)$$

where $\mathbf{W}_h^Q$ and $\mathbf{W}_h^K$ denote the projection matrices for the h -th attention head, and d_k is the hidden dimension of each head.

To integrate data-driven attention scores with problem-specific geometric priors, we introduce a proximity heuristic. The basic idea is that UAV paths tend to connect spatially nearby nodes, so scoring should explicitly favor closer candidates. Accordingly, the adjusted attention score for the j -th candidate node is defined as:

$$S_j = s_j[t] - \text{Geo}_j \quad (32)$$

where Geo_j represents the cost value from the current node to candidate node j , typically taken as the Euclidean distance, but it can also be extended to include energy consumption, flight time, or other task-related cost functions. By adjusting

the attention scores in this manner, even if the data-driven attention mechanism tends to favor distant nodes, the proximity heuristic corrects this bias, ensuring that the generated path is geometrically reasonable.

After obtaining the adjusted scores, we calculate the selection probability $p_j[t]$, which represents the probability that the current UAV selects the j -th node as the next hovering point to collect data in the t -th time step:

$$p_j[t] = \text{Softmax}(C \cdot \tanh(S_j)) \mathbf{M}_j \quad (33)$$

where C is a scaling factor and $\mathbf{M}$ is the masking. Initially, we set $\mathbf{M} = [1, \dots, 1]$ at the beginning of path planning. The length of $\mathbf{M}$ is N . When $\mathbf{M}_i = 1$ means the UAV can select the i -th node. There are three cases when $\mathbf{M}_i = 0$: (1) The i -th node has already been visited; (2) the UAV cannot fly to the position of the i -th node due to insufficient battery power; and (3) the UAV cannot collect data from the i -th node due to limited storage capacity. At this point, Problem $P2$ has been solved.

Then, since Problem $P1$ is a convex optimization problem. The optimal data collection time $\mathcal{T}^C$ and energy transfer time $\mathcal{T}^E$ are achieved when Eq. (9) is satisfied with equality as follows:

$$T_{ij}^C[t]^* B \log_2 \left(1 + \frac{h_{ij}[t] E_{ij}^R[t]^*}{\sigma^2 T_{ij}^C[t]^*} \right) = D_i, \forall j \in \mathcal{M}, \forall i \in \mathcal{N}. \quad (34)$$

The convex optimization problem can be solved by applying the Karush–Kuhn–Tucker (KKT) conditions [28], and the corresponding optimal solutions $\{T_{ij}^C[t]^*, T_{ij}^E[t]^*\}$ are then derived following the analytical procedure in [20]. Hence, we derive the optimal energy transfer time and data collection time using the KKT conditions, based on which the Problem $P1$ is solved.

E. Complexity Analysis

Let n denote the number of hovering points, d the hidden dimension, and L the number of encoder layers. The computational complexity of the proposed SLM is mainly dominated by the BERT encoder and the path decoder. Specifically, the complexity of each encoder layer is $O(n^2d + nd^2)$, and thus the total complexity of the encoder is $O(L(n^2d + nd^2))$. The path decoder needs to score all candidate nodes while sequentially generating a complete path, resulting in a complexity of $O(n^2d)$. By contrast, the enhanced embedding and contextual feature construction only involve linear projection and aggregation, with complexity $O(nd)$, which is negligible compared with the encoder and decoder. Therefore, the overall single-view inference complexity of the SLM can be expressed as $O(L(n^2d + nd^2) + n^2d)$.

F. Ensemble Inference

During the inference phase, we leverage the symmetry of the D_4 group to further enhance solution quality and robustness. For each geometrically equivalent transformation $g \in \mathcal{G}$, a new instance is generated under the transformation and decoded independently to obtain the path π_g . The path π_g is then

mapped back to the original coordinate system via the inverse transformation $g^{-1}(\pi_g)$. Finally, multiple solutions obtained under different equivalent coordinate systems are compared, and the optimal one is selected as the final output:

$$\pi^* = \arg \min_{g \in \mathcal{G}} L(g^{-1}(\pi_g)) \quad (35)$$

where L denotes the objective function, which in this work corresponds to Problem P2. Compared to single-view inference, this approach significantly reduces the risk of suboptimal solutions caused by model bias or numerical perturbations, thereby improving overall stability and reliability. Moreover, ensemble inference ensures consistent model performance across different coordinate orientations, substantially enhancing cross-distribution generalization and preventing the model from being trapped in local optima due to attention biases under a specific geometric arrangement.

G. Reinforcement Path Optimization

1) *Policy Gradient Update*: During the training phase, we adopt a policy gradient-based reinforcement learning framework to optimize the path quality generated by the path decoder in an end-to-end manner. Specifically, the model parameters θ are optimized to minimize the expected path length. To further improve training efficiency and leverage the geometric symmetry inherent in path planning, for a given instance, paths are decoded simultaneously from multiple different starting points, resulting in a set of equivalent complete paths. Let the path set obtained for instance $\mathcal{I}_i$ from N different starting points be $\{\tau_{ij}\}_{j=1}^N$. The returns of all paths are then used to update the policy gradient:

$$\nabla_{\theta} J(\theta) \approx \frac{1}{BN} \sum_{i=1}^B \sum_{j=1}^N \hat{R}(\tau_{ij}) \nabla_{\theta} \log p_{\theta}(\tau_{ij} | \mathcal{I}_i) \quad (36)$$

where B denotes the batch size, τ_{ij} represents the i -th sampled path sequence for instance $\mathcal{I}_i$, and $\hat{R}(\tau_{ij})$ is the normalized reward. Moreover, since the path length distributions can vary significantly across different instances, directly using absolute path length as the reward may lead to large variations in cross-instance gradient updates. To address this, rewards are normalized using the mean and standard deviation within each instance:

$$\hat{R}(\tau_{ij}) = \frac{R(\tau_{ij}) - \mu(\mathcal{I}_i)}{\sigma(\mathcal{I}_i)} \quad (37)$$

where $\mu(\mathcal{I}_i)$ and $\sigma(\mathcal{I}_i)$ denote the mean and standard deviation of the rewards across all equivalent paths for instance $\mathcal{I}_i$. Theoretically, reward normalization can effectively reduce the variance of policy gradient estimation across samples with different reward scales, thereby mitigating optimization oscillation and training instability caused by inconsistent return distributions. Furthermore, by mapping the rewards of different instances into a comparable numerical range, this mechanism provides smoother and more consistent learning signals for policy updates, thus improving training stability and accelerating convergence [29].

Additionally, during training, a subset of paths from the set $\{\tau_{ij}\}_{j=1}^N$ is randomly sampled and subjected to D_4 group-equivalent transformations, generating samples that are geometrically equivalent to the original problem but have different distributions. This not only significantly increases the diversity of the training data but also prevents the SLM model from overfitting to specific coordinate systems or regional distributions, enabling it to learn geometry-invariant features.

2) *Reward Function Design*: During training, the optimization objective of the model parameters θ is equivalent to maximizing the expected return of the negative path length. The reward function is thus defined as the negative of the path length. In addition, a penalty term is introduced based on the path length to assign negative rewards to solutions that violate constraints:

$$R(\tau) = -L_{\theta}(\tau) - \lambda \cdot (\Phi_E(\tau) + \Phi_C(\tau)) \quad (38)$$

where $L_{\theta}(\tau)$ denotes the distance function, $\lambda > 0$ is the penalty coefficient, $\Phi_E(\tau)$ quantifies the degree of violation of the energy constraint for path τ , and $\Phi_C(\tau)$ quantifies the degree of violation of the storage constraint for path τ .

For the energy constraint, it is defined as

$$\Phi_E(\tau) = \sum_{k=1}^K \max(0, E_k(\tau) - E_k^{\max}) \quad (39)$$

where $E_k(\tau)$ denotes the energy consumption of UAV k along path τ , and $E_k^{\max}$ is its maximum battery capacity.

Similarly, for the storage constraint, it is formulated as

$$\Phi_C(\tau) = \sum_{k=1}^K \max(0, C_k(\tau) - C_k^{\max}) \quad (40)$$

where $C_k(\tau)$ represents the storage requirement of UAV k along path τ , and $C_k^{\max}$ denotes its maximum storage capacity.

V. SOLUTION FRAMEWORK FROM LLM TO AGENTIC AI

A. LLM-Based Solution Framework

Multimodal Large Language Models (MLLMs) demonstrate unique advantages in energy-efficient UAV-enabled WPT systems. Traditional methods rely on numerical coordinates and distance matrices to optimize paths, whereas MLLMs can reason directly from multimodal inputs such as images and text, leveraging visualized point distribution information to generate feasible paths. This capability emulates human spatial intuition and visual reasoning, avoiding complex numerical computations. Moreover, MLLMs exhibit strong generalization in zero-shot and few-shot scenarios, enabling rapid generation of near-optimal solutions without task-specific training data [30].

We design an MLLM-based solution framework that employs visual reasoning and chain-of-thought reasoning to simultaneously solve the following two problems: (i) optimization of energy transfer and data collection time, and (ii) optimization of UAV flight path. First, structured prompts are constructed to explicitly inject the system model, parameters, and constraints into the MLLM, together with the visualized distribution of UAV hovering points. By reasoning directly

over these multimodal representations, the MLLM can generate a feasible flight trajectory that attempts to satisfy the given constraints (defined as Problem $P2$). Subsequently, continuous variable optimization is performed along this trajectory. Although most MLLMs cannot explicitly derive the KKT conditions, they are capable of generating executable Python code for solving them. By invoking an external code interpreter to run this generated code, we further obtain the optimal energy transfer time and data collection time, corresponding to Problem $P1$. The proposed MLLM-based solution framework is illustrated in Fig. 3.

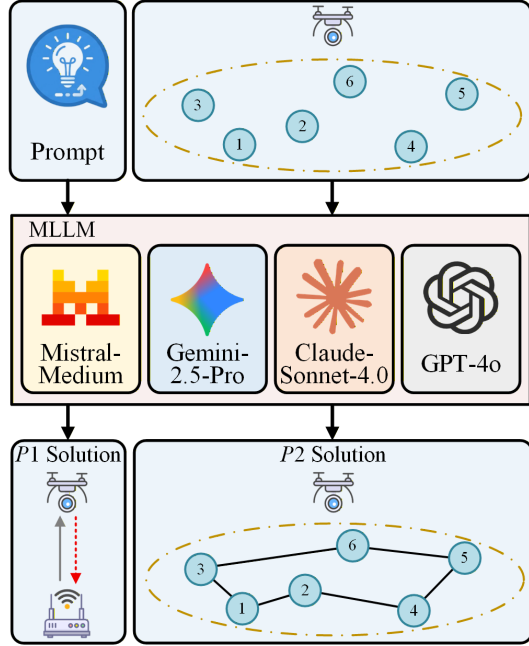


Figure 3: The MLLM-Based Solution Framework.

B. Agentic AI-Based Solution Framework

Through multi-agent collaborative iterative optimization, MLLMs can effectively reduce path intersections, improve coverage completeness, and enhance solution robustness [31]. These features make MLLMs a powerful tool for UAV path planning in complex environments with multiple constraints.

Based on these insights, we propose an Agentic AI system for solving UAV path planning problems. The core idea is to integrate MLLMs with the MCP to achieve intelligent path planning for UAV networks. The system comprises four types of agents: Initializer, Actor, Critic, and Reflector, which collaboratively form a closed-loop process of “generation–optimization–evaluation–reflection” for UAV paths. The proposed Agentic AI system is illustrated in Fig. 4.

Specifically, the Initializer generates feasible initial path solutions, while the Actor produces diverse candidate paths via high-temperature sampling. The Critic evaluates the paths, invoking the MCP module to verify constraint satisfaction and provide scores for each candidate. The Reflector generates targeted improvement suggestions based on the scoring results, which are fed back to the Initializer, thereby guiding subsequent iterations in a more focused search direction.

This framework significantly enhances system autonomy, interpretability, and constraint feasibility. The workflow of the Agentic AI system is summarized as follows:

1) *Initializer for Initial Solution Generation*: The goal of Initializer is to construct a feasible initial path for the UAV system. The input consists of a map containing a data center and N hovering nodes. This agent partitions the nodes such that each UAV has a closed path starting and ending at the data center, ensuring that every node is visited exactly once. Moreover, during iterative optimization, the Initializer leverages suggestions from the Reflector to make targeted adjustments, minimizing path redundancy and intersections to reduce the difficulty of subsequent optimization.

2) *Actor for Candidate Solution Optimization*: The Actor is responsible for improving upon the initial solutions generated by the Initializer. By employing high-temperature sampling (temperature = 0.7), the Actor can generate multiple candidate path planning solutions in a single iteration, thereby increasing the diversity of the solution space. Its optimization objectives include minimizing path length, balancing task load, and ensuring geometric feasibility of the paths.

3) *Critic for Candidate Solution Evaluation*: The Critic evaluates the quality of the solutions generated by the Actor. The performance of each candidate solution is decomposed into five dimensions: coverage completeness, path closure, geometric rationality, task balance, and constraint satisfaction. Constraint satisfaction is verified through the MCP protocol. Specifically, the MCP module invokes the system model code to rigorously check energy and storage constraints for each candidate path, returning the results to the Critic, which then computes individual scores and a weighted overall score for each solution.

4) *Reflector for Reflective Analysis and Improvement*: The Reflector is responsible for analyzing the scoring results provided by the Critic, selecting the current best path, diagnosing it, and generating targeted improvement suggestions. Its analysis scope includes filling in uncovered nodes, repairing path closure, reducing geometric intersections, rebalancing task loads, and adjusting for constraint violations. The current best path along with the improvement recommendations is then sent back to the Initializer, guiding the next iteration of the optimization process.

5) *Energy transfer and Data Collection Time Optimization*: Upon obtaining the optimized path, and under prompt guidance, the MCP proceeds to invoke additional optimization tools, such as KKT-based or Genetic Algorithm (GA) solvers, to optimize the energy transfer time and data collection time, thereby solving Problem $P1$.

In the proposed Agentic AI framework, the interactions among agents are primarily guided by the visual reasoning capability of the MLLM, enabling multimodal perception and reflective collaboration for path generation and optimization. When the MLLM produces suboptimal planning results—such as path intersections or incomplete coverage—the Agentic AI system can invoke external local search algorithms (e.g., 2-opt or 3-opt) via the MCP for further refinement. This hybrid mechanism combines the global visual reasoning of

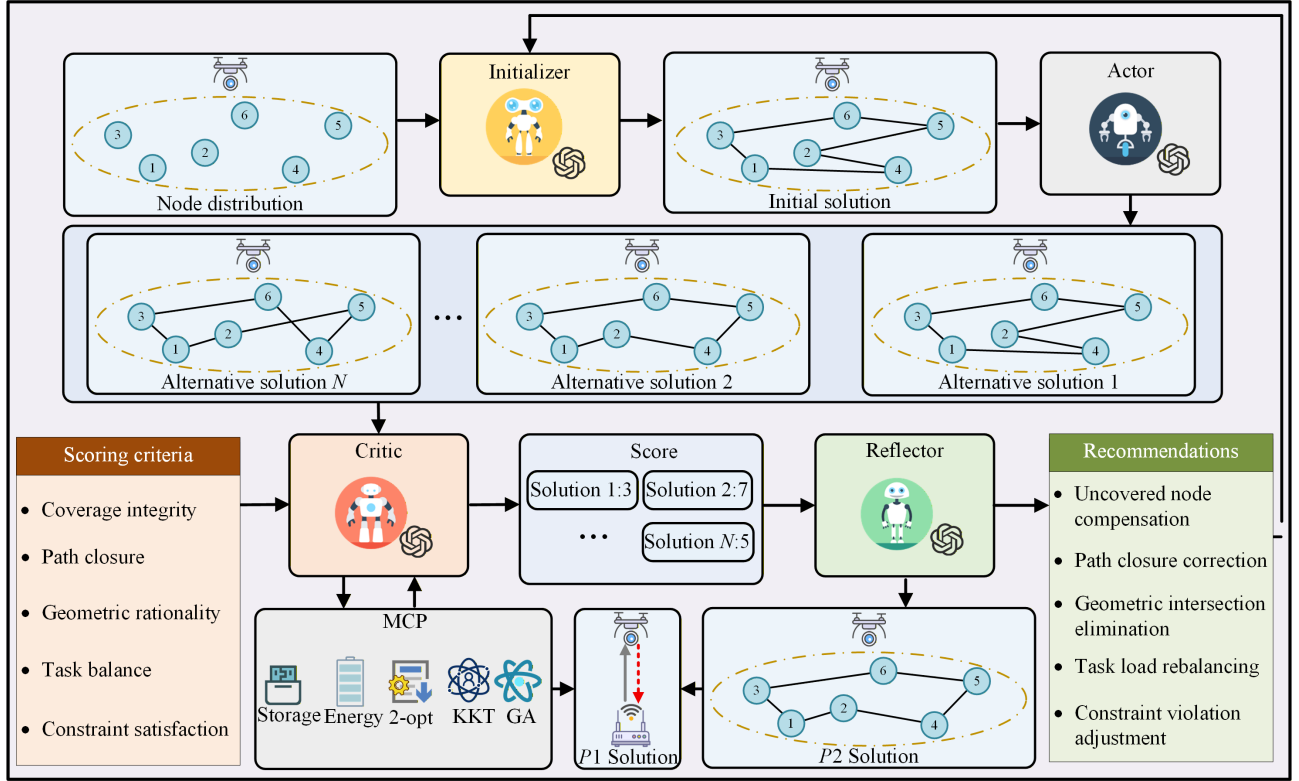


Figure 4: The Agentic AI-Based Solution Framework.

the MLLM with the local optimization strength of heuristic search, significantly improving path optimality.

VI. RESULTS AND DISCUSSION

A. Simulation Settings

All simulations are conducted in a $1000 \text{ m} \times 1000 \text{ m}$ square area to emulate a typical multi-UAV data collection scenario. In the area, 100 latent cluster centers are first randomly generated, and then 500 IoTs are randomly distributed around these centers. The data size D_i for each IoT is randomly selected within the range of $[0.2, 0.5]$ MB. The data center is located at the coordinate $(500, 500)$. All UAVs depart from the data center to perform their missions and return to the same location after completing a full flight cycle. Each UAV operates at a fixed altitude of $H_z = 20 \text{ m}$. Other detailed parameters of the system model are summarized in Table II. These settings are adopted as a baseline scenario for fair comparison. In particular, the LS-FCM algorithm is applied to determine the final UAV hovering points, the data center can be deployed at arbitrary locations, and the formulation can be further extended to heterogeneous UAV-capacity scenarios.

In the SLM scheme, the DistilBERT model is adopted, consisting of 6 Transformer encoder layers, 12 self-attention heads, and a hidden dimension of 768, resulting in approximately 66 million parameters [32]. To adapt DistilBERT to UAV path planning, we fine-tune the model by introducing an enhanced UAV embedding layer (positional and geometric-angle features), designing an 8-head path decoder.

In the LLM and Agentic AI schemes, GPT-4o is utilized. Each agent adopts a temperature-controlled sampling mecha-

nism (Temperature = 0.7) to balance exploration and stability during reasoning.

Table II: Parameters of the System Model.

Parameter	Value
UAV Data Storage Capacity $C_{\max}$	50 MB
UAV flight speed v	10 m/s
Flight Height H_z	20 m
UAV Battery Capacity $E_{\max}$	2550 mAh
Bandwidth B	2 MHz
Transmit power P^T	10 W
Flight power P^F	75 W
Hover power P^H	50 W
Signal attenuation factor μ	0.8
LoS probability factor B_1, B_2, B_3, B_4	-0.45, 0.04, -0.63, 1.63
Channel power gain β_0	-30 dB
Noise power σ^2	-110 dB
Attenuation parameter η^L	0.9
Average path loss exponents α_L, α_N	2.5, 3.5

The entire simulation framework is implemented in Python 3.8 using the PyTorch 1.13.0 deep learning library. All experiments are conducted on a workstation equipped with an Intel Xeon CPU and an NVIDIA A800 GPU with 80 GB of memory. For SLM training, LoRA-based fine-tuning is adopted, with the total number of training epochs set to 50 and the batch size set to 64. The reinforcement learning training dataset contains 1 million samples, and the minimum learning rate is set to 1×10^{-5} .

B. Experiment 1: Ablation Study of the SLM

This experiment aims to systematically evaluate the incremental contribution of each structural component in the proposed SLM. The number of UAV hovering points is defined as 100, and a single UAV is assumed to visit all hovering points. To this end, five model variants are designed, where each variant progressively incorporates one additional module based on the previous configuration:

- **SLM-Base:** The baseline model adopts a BERT with routing decoder [33].
- **SLM-E:** The enhanced UAV embedding layer is introduced to encode geometric positional information.
- **SLM-C:** Building upon SLM-E, this variant further integrates an enhanced contextual feature query mechanism.
- **SLM-G:** The geometry-aware path decoder is incorporated.
- **SLM (Ours):** The complete version integrates all above modules and the ensemble inference.

All model variants are trained under identical configurations using the reinforcement path optimization framework. Their performance is evaluated from three perspectives: the total energy cost E_{tot} , the trajectory length L_{traj} , and the optimality gap G_{opt} . We define the optimality gap G_{opt} as the relative sub-optimality with respect to the best-known solution:

$$G_{\text{opt}} = \frac{L_{\text{traj}} - L_{\text{traj}}^*}{L_{\text{traj}}^*} \times 100\% \quad (41)$$

where L_{traj}^* denotes the minimum trajectory length among all compared methods; $G_{\text{opt}} = 0$ indicates that the current solution attains the best performance within the set of considered methods. The comparison results of SLM variants, together with the Optimal solution (OPT) and the conventional 2-opt baseline, are listed in Table III.

Table III: Comparison of SLM Variants with OPT and 2-opt.

Model Variant	E_{tot} (Wh)	L_{traj} (m)	G_{opt} (%)
OPT	-	7760.00	0
2-opt	38.98	9072.67	16.91
SLM-Base	36.72	7986.43	2.91
SLM-E	36.45	7856.95	1.25
SLM-C	36.38	7832.38	0.93
SLM-G	36.31	7798.46	0.49
SLM (Ours)	36.22	7773.29	0.17

The results show that each enhancement progressively improves performance. The full SLM achieves the minimum energy consumption and yields the shortest flight path. This improvement stems from three factors: (1) enhanced embeddings provide richer geometric cues, (2) contextual queries capture both global and historical dependencies, and (3) the path decoder grounded in geometric knowledge and ensemble inference introduces spatial symmetry awareness, thereby enhancing generalization to unseen UAV layouts.

C. Experiment 2: Comparison with Other Path Planners

To verify the superiority of the proposed SLM, we compare it with three lightweight path planning models: ATOM [2],

Pointerformer [34], and AttRouting [33]. The number of UAV hovering points is defined as 100, and a single UAV is assumed to visit all hovering points. All methods employ the same encoder architecture to ensure comparable parameter scales. The models are trained on identical datasets and evaluated in terms of the flight energy cost E^F , the standard deviation StD and the training time T_{tra} . The performances of all path planning models are listed in Table IV.

Table IV: Comparison with Different Path Planners.

Method	E^F (Wh)	StD	T_{tra} (h)	T_{inf} (s)
ATOM	16.42	0.35	8.56	3.57
Pointerformer	16.33	0.12	7.89	4.02
AttRouting	17.25	0.43	10.43	2.64
SLM (Ours)	16.19	0.08	6.85	4.27

The proposed SLM attains the lowest flight energy consumption and the smallest standard deviation. This is because its geometry-aware path decoder effectively learns spatial regularities, while enhanced contextual queries and ensemble inference yield more stable policy gradients. In addition, SLM also achieves the fastest training speed, because it is fine-tuned from a pretrained BERT backbone and thus benefits from the pretrained Transformer's structural and optimization priors, whereas the other models are trained from scratch. These priors improve training stability and accelerate convergence. Among these path planners, SLM incurs the longest inference time, mainly because its geometric feature enhancement and ensemble inference introduce additional computational overhead during the inference stage. By contrast, ATOM and Pointerformer can also reach near-optimal solutions but require substantially longer training time.

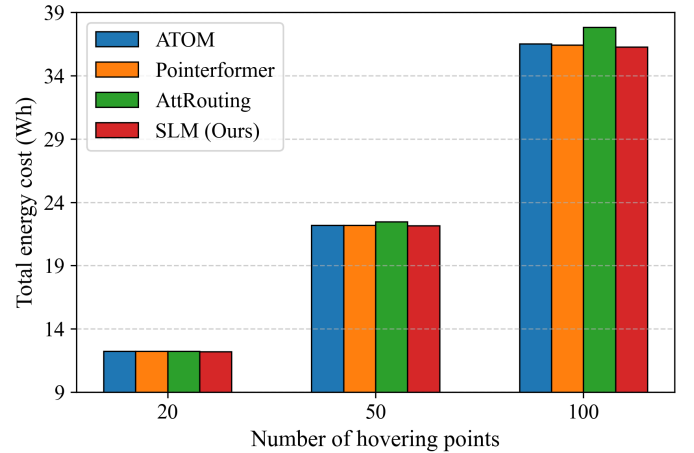


Figure 5: Energy Comparison of Lightweight Path Planners under Different Numbers of Hovering Points.

We further compare the variations in total energy cost of different planners under varying numbers of hovering points. To ensure a fair comparison, the node distribution density is kept identical across all simulation scenarios. Fig. 5 illustrates that the total energy cost of all planners increases with the number of hovering points. SLM consistently achieves the lowest energy consumption, while AttRouting yields the

highest, and their gap widens as the problem scale grows, demonstrating SLM’s superior scalability.

D. Experiment 3: Comparison of Different MLLMs

To evaluate the reasoning capabilities of MLLMs for multi-UAV path planning tasks, we compare four representative models: GPT-4o, Gemini-2.5-Pro, Claude-Sonnet-4.0, and Mistral-Medium. Each model reasons over visualized coordinate maps to generate trajectories, followed by interactive refinement through human-in-the-loop prompting. The simulation area is set to 550 m × 550 m, containing 150 IoT devices, and the number of UAV hovering points is defined as 30. The evaluation metrics include the total energy cost E_{tot} , the flight energy consumption E^F , and the inference time T_{inf} . The performances of all MLLMs are listed in Table V.

Table V: Performance Comparison of Different MLLMs via Visual Reasoning.

Model	E_{tot} (Wh)	E^F (Wh)	T_{inf} (s)
Mistral-Medium	15.86	9.85	8.02
Gemini-2.5-Pro	13.66	7.64	55.25
Claude-Sonnet-4.0	14.57	8.58	9.63
GPT-4o	13.71	7.71	8.34

Experimental results show that among all MLLMs, Gemini-2.5-Pro demonstrates stronger visual reasoning and achieves lower energy consumption, albeit with a substantially longer inference time than its counterparts. GPT-4o delivers competitive performance with lower inference latency.

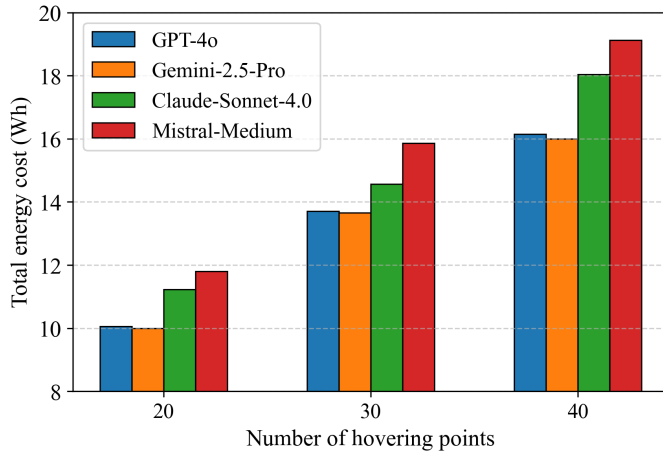


Figure 6: Energy Cost across MLLMs under Different Hovering Node Scales.

Fig. 6 compares the total energy consumption of four MLLMs under different numbers of hovering nodes. As the number of nodes increases, the energy consumption of all models rises due to the growth in trajectory length. Among them, Gemini-2.5-Pro consistently achieves the lowest energy consumption, reflecting its stronger visual reasoning and trajectory optimization capabilities, while GPT-4o exhibits more balanced overall performance.

E. Experiment 4: Multi-Agent Collaboration Effectiveness

This experiment aims to investigate the role of multi-agent collaboration within the Agentic AI framework for multi-UAV path planning, where GPT-4o is adopted as the underlying MLLM. Five configurations are evaluated: (A1) a single-agent system based on GPT-4o that autonomously refines trajectories through a ReAct mechanism [35], (A2) Initializer + Actor, (A3) Initializer + Actor + Critic, (A4) Initializer + Critic + Reflector, and (A5) the full framework incorporating four agents. The simulation area is set to 550 m × 550 m, containing 150 IoT devices, and the number of UAV hovering points is defined as 30. All systems perform iterative optimization under the same experimental environment and are compared in terms of the total energy cost E_{tot} , the constraint satisfaction rate R_{sat} , and the inference time T_{inf} . We define the constraint satisfaction rate R_{sat} as the fraction of planned paths that satisfy all feasibility constraints. The performances of all Agent configurations are listed in Table VI.

Table VI: Performance Comparison of Different Agent Configurations.

Configuration	E_{tot} (Wh)	R_{sat} (%)	T_{inf} (s)
A1: Single-agent + ReAct	13.68	20	74.52
A2: Initializer + Actor	12.95	27	138.26
A3: Initializer + Actor + Critic	12.72	72	105.25
A4: Initializer + Critic + Reflector	12.70	78	187.85
A5: Four-Agent	12.43	83	116.47

Experimental results show that the complete Agentic AI framework (A5) delivers the best overall performance: compared with other configurations, it achieves the highest constraint satisfaction rate and lower energy consumption. Specifically, the Critic introduces constraint-aware feedback via the MCP protocol to ensure solution feasibility; the Reflector provides reflective guidance that dynamically corrects suboptimal trajectories. The collaboration among these agents establishes an adaptive “generation-optimization-evaluation-reflection” loop, thereby enhancing the Agentic system’s intelligence, stability, and interpretability.

To overcome the limitations of pure visual reasoning and further enhance the performance of the Agentic AI system, we introduce a new agent configuration, A6, in which the Critic incorporates a 2-opt local search tool via the MCP Protocol. Fig. 7 compares the total energy consumption of different Agent configurations with 30, 40, and 50 hovering points. As the problem scale increases, all configurations exhibit a steady rise in energy consumption. The complete Agentic AI integrated with 2-opt local search consistently achieves the lowest energy cost across all scales, demonstrating the significant effectiveness of the MCP + tool mechanism in compensating for the limitations of visual reasoning and enhancing trajectory optimization performance.

F. Experiment 5: Comparison among SLM, LLM, and Agentic AI

A unified comparison is conducted across the following paradigms under varying problem scales in multi-UAV path

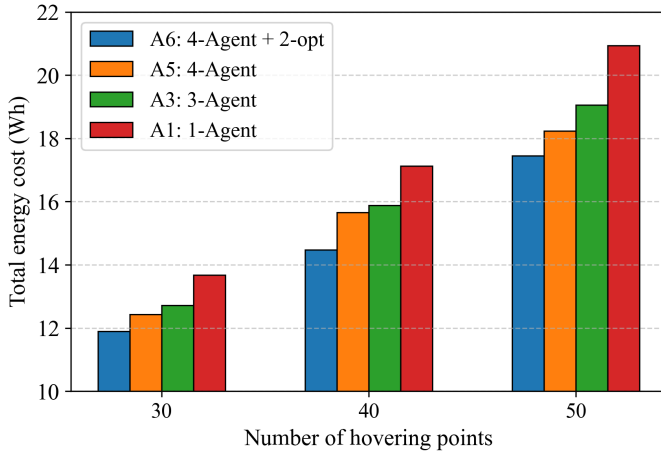


Figure 7: Energy Comparison of Agent Configurations under Different Numbers of Hovering Points.

planning scenarios: SLM, MLLM (GPT-4o), and Agentic AI. Each method is evaluated based on the trajectory length L_{traj} , the total energy cost E_{tot} , the constraint satisfaction rate R_{sat} , and the inference time T_{inf} . Tables VII and VIII report the performance comparison among SLM, LLM, and Agentic AI systems for 20 and 100 hovering points.

Table VII: Comparison among SLM, LLM, and Agentic AI Systems with 20 Nodes.

Method	L_{traj} (m)	E_{tot} (Wh)	R_{sat} (%)	T_{inf} (s)
SLM	5428.75	15.56	95	3.48
MLLM (GPT-4o)	5685.52	16.09	33	8.62
Agentic AI	5130.44	14.94	85	87.85
Agentic AI + 2-opt	4540.12	13.71	85	112.47

Table VIII: Comparison among SLM, LLM, and Agentic AI Systems with 100 Nodes.

Method	L_{traj} (m)	E_{tot} (Wh)	R_{sat} (%)	T_{inf} (s)
SLM	8938.45	39.72	92	4.35
MLLM (GPT-4o)	10865.26	42.72	12	9.56
Agentic AI	10610.21	42.18	68	384.65
Agentic AI + 2-opt	9468.36	39.83	68	443.27

The results indicate that, in small-scale system optimization, both SLM and Agentic AI achieve low energy consumption; in particular, the Agentic AI system augmented with 2-opt local search attains performance that is nearly at the theoretical optimum. Because constraint information is incorporated during training, SLM sustains the highest constraint satisfaction rate, and its inference latency is extremely low—substantially lower than that of MLLMs and Agentic AI systems. In large-scale system optimization, the performance of vision-reasoning-based MLLMs and Agentic AI degrades markedly, whereas SLM delivers the best overall results, achieving the highest constraint satisfaction rate and the lowest inference latency. Although integrating 2-opt into the Agentic AI pipeline

can compensate for the limitations of visual reasoning and yield competitive performance, it comes at the cost of a significant increase in inference latency.

G. Summary and Lessons Learned

Under comprehensive and rigorous evaluation, we summarize the characteristics and applicability of SLM, LLM, and Agentic AI for energy-efficient UAV-enabled WPT systems as follows, with the overall comparison illustrated in Fig. 8.

1) *SLM*: Given comparable hardware and task scales, SLM attains the highest computational efficiency and the lowest inference latency, and achieves the best energy performance even in large-scale system optimization. Because constraint knowledge is explicitly injected during training, SLM typically yields the highest constraint satisfaction rate and more stable policy gradients. Its limitation lies in task specificity and limited cross-task generalization, necessitating retraining for different optimization objectives or constraint sets.

Applicability: real-time, resource-constrained deployments across varying problem scales, or scenarios with stable objectives/constraints that allow offline retraining.

2) *LLM*: LLMs exhibit strong zero-shot reasoning and generalization in small-scale optimization problems, enabling rapid validation and transfer across tasks. However, for large-scale optimization problems (e.g., path planning), they incur substantial computational overhead and higher latency, with degraded accuracy—reflecting current bottlenecks in visual reasoning and long-horizon global optimization.

Applicability: small-scale, exploratory tasks requiring quick adoption and cross-domain generalization; less suitable for stringent real-time constraints and severely limited compute.

3) *Agentic AI*: Agentic AI offers a well-balanced overall profile by enabling self-correction and adaptive planning via the “generation-optimization-evaluation-reflection” loop. In large-scale optimization problems, integrating external tools via MCP (e.g., 2-opt and KKT) can markedly compensate for MLLM’s limitations in visual reasoning, driving the energy/trajectory toward near-optimality—at the cost of increased inference latency and more optimization steps.

Applicability: medium-to-large problem sizes where feasibility, robustness, and interpretability are prioritized, and where some latency can be traded for higher solution quality.

VII. CONCLUSIONS

This study investigates the advantages and applicability of SLMs, LLMs, and Agentic AI for UAV-enabled WPT systems. First, we propose a lightweight BERT-based framework with enhanced UAV embeddings, contextual features, a geometry-aware path decoder, and ensemble inference for efficient path generation and time allocation under limited computation and energy resources. Second, to address dynamic environments and multiple constraints, we develop an Agentic AI framework to exploit the reasoning and interactive capabilities of LLMs. Through a closed-loop process of path generation, optimization, evaluation, and reflection, the system enables more adaptive and intelligent decision-making. Experimental results show that SLMs achieve low latency and high energy

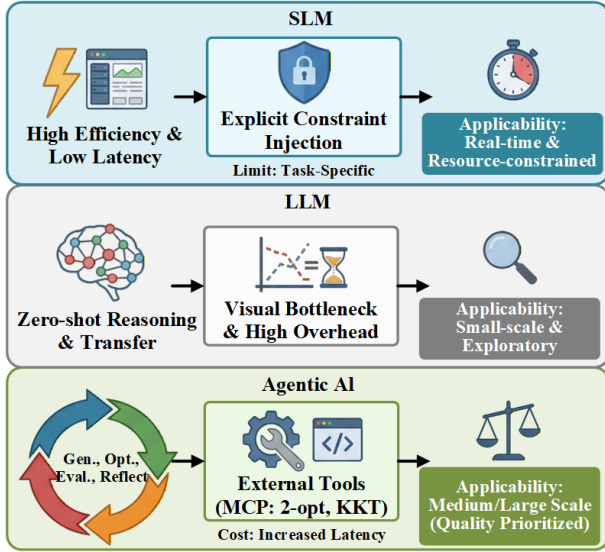


Figure 8: Comparison of SLM, LLM, and Agentic AI Paradigms.

efficiency in resource-constrained scenarios, whereas LLMs and Agentic AI provide stronger generalization and decision-making capability in cross-task planning.

Future work can further enhance UAV-enabled WPT systems in several directions. First, model lightweighting and compression can reduce deployment overhead on edge or airborne platforms. Techniques such as model distillation, lightweight LLMs (e.g., LLaMA2-7B), and edge-cloud collaboration can further lower inference cost. Second, expanding tool-chain integration in Agentic AI frameworks may improve optimization performance. Third, stronger cross-scenario generalization and extension to mobile IoT scenarios will improve adaptation to more dynamic UAV-enabled WPT environments. Finally, practical engineering issues, including communication latency, on-device inference power, privacy, and security, should be addressed to strengthen real-world applicability.

REFERENCES

- [1] S. Wang, X. Zhang, Z. Song, Z. Hua, J. Song, J. Wang, G. Pan, and D. Niyato, "Generative ai empowered covert communications: Autonomy, efficiency, and suitability," *IEEE Wireless Communications*, pp. 1–9, 2025.
- [2] L. Dong, F. Jiang, and Y. Peng, "Attention-based uav trajectory optimization for wireless power transfer-assisted iot systems," *IEEE Transactions on Industrial Electronics*, vol. 72, no. 8, pp. 8463–8471, 2025.
- [3] C. S. Tan, R. Mohd-Mokhtar, and M. R. Arshad, "A comprehensive review of coverage path planning in robotics using classical and heuristic algorithms," *IEEE Access*, vol. 9, pp. 119 310–119 342, 2021.
- [4] M. Sil, S. Bhaumik, and R. K. Barai, "A convex optimization based path planning algorithm for robot motion in constrained space," in *2020 IEEE Applied Signal Processing Conference (ASPCON)*, 2020, pp. 188–192.
- [5] Z. Ni, Z. Hua, X. Yang, R. Zhang, S. Wang, and G. Pan, "Llm-aided spectrum-sharing leo satellite communications," *IEEE Journal on Selected Areas in Communications*, vol. 44, pp. 2591–2605, 2026.
- [6] F. Jiang, C. Pan, L. Dong, K. Wang, M. Debbah, D. Niyato, and Z. Han, "A comprehensive survey of large ai models for future communications: Foundations, applications, and challenges," *IEEE Communications Surveys & Tutorials*, vol. 28, pp. 4731–4764, 2026.
- [7] Y. Yang and X. Liu, "Deep reinforcement learning based trajectory optimization for uav-enabled iot with swipt," *Ad Hoc Networks*, vol. 159, p. 103488, 2024.
- [8] L. Dong, F. Jiang, and Y. Peng, "Attention-based uav trajectory optimization for wireless power transfer-assisted iot systems," *IEEE Transactions on Industrial Electronics*, 2025.
- [9] S. Muy, V. That, and J.-R. Lee, "Trajectory-aware uav-enabled wpt networks based grid world drl approach," in *2024 International Conference on Information Networking (ICOIN)*. IEEE, 2024, pp. 440–445.
- [10] C. Kim, H.-H. Choi, and K. Lee, "Joint optimization of trajectory and resource allocation for multi-uav-enabled wireless-powered communication networks," *IEEE Transactions on Communications*, vol. 72, no. 9, pp. 5752–5764, 2024.
- [11] Z. Ning, Y. Yang, X. Wang, Q. Song, L. Guo, and A. Jamalipour, "Multi-agent deep reinforcement learning based uav trajectory optimization for differentiated services," *IEEE Transactions on Mobile Computing*, vol. 23, no. 5, pp. 5818–5834, 2024.
- [12] G. Su, "Deep reinforcement learning aided trajectory and power control for secure uav communication," in *Proceedings of the 2024 11th International Conference on Wireless Communication and Sensor Networks*, 2024, pp. 74–79.
- [13] D. Li, B. Du, and Z. Bai, "Deep reinforcement learning-enabled trajectory and bandwidth allocation optimization for uav-assisted integrated sensing and covert communication," *Drones*, vol. 9, no. 3, p. 160, 2025.
- [14] H. Samma and S. El-Ferik, "Uav visual path planning using large language models," *Transportation Research Procedia*, vol. 84, pp. 339–345, 2025.
- [15] F. Jiang, L. Dong, X. Pan, K. Wang, and C. Pan, "Agentic ai empowered multi-uav trajectory optimization in low-altitude economy networks," *arXiv preprint arXiv:2508.16379*, 2025.
- [16] B. Yin, X. Fang, X. Wang, L. Yan, J. Wu, and J. Wang, "Trajectory design and beamforming in uav-assisted wireless networks: A fine-tuned m2llm-driven drl-based framework," *IEEE Transactions on Wireless Communications*, 2025.
- [17] C. You and R. Zhang, "Hybrid offline-online design for uav-enabled data harvesting in probabilistic los channels," *IEEE Transactions on Wireless Communications*, vol. 19, no. 6, pp. 3753–3768, 2020.
- [18] A. Al-Hourani, S. Kandeepan, and S. Lardner, "Optimal lap altitude for maximum coverage," *IEEE Wireless Communications Letters*, vol. 3, no. 6, pp. 569–572, 2014.
- [19] G. Pan, H. Zhang, R. Zhang, S. Wang, J. An, and M.-S. Alouini, "Space simultaneous information and power transfer: An enhanced technology for miniaturized satellite systems," *IEEE Wireless Communications*, vol. 30, no. 2, pp. 122–129, 2023.
- [20] H. Hu, K. Xiong, G. Qu, Q. Ni, P. Fan, and K. B. Letaief, "Aoi-minimal trajectory planning and data collection in uav-assisted wireless powered iot networks," *IEEE Internet of Things Journal*, vol. 8, no. 2, pp. 1211–1223, 2020.
- [21] T. Wu, J. Liu, J. Liu, Z. Huang, H. Wu, C. Zhang, B. Bai, and G. Zhang, "A novel ai-based framework for aoi-optimal trajectory planning in uav-assisted wireless sensor networks," *IEEE Transactions on Wireless Communications*, vol. 21, no. 4, pp. 2462–2475, 2021.
- [22] F. Jiang, K. Wang, L. Dong, C. Pan, W. Xu, and K. Yang, "Deep learning based joint resource scheduling algorithms for hybrid mec networks," 2019.
- [23] D. Chen, Q. Qi, Q. Fu, J. Wang, J. Liao, and Z. Han, "Transformer-based reinforcement learning for scalable multi-uav area coverage," *IEEE Transactions on Intelligent Transportation Systems*, vol. 25, no. 8, pp. 10 062–10 077, 2024.
- [24] P. Sharma, "Dihedral group d4—a new feature extraction algorithm," *Symmetry*, vol. 12, no. 4, p. 548, 2020.
- [25] M. M. Bronstein, J. Bruna, T. Cohen, and P. Velicković, "Geometric deep learning: Grids, groups, graphs, geodesics, and gauges," *arXiv preprint arXiv:2104.13478*, 2021.
- [26] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, 2019, pp. 4171–4186.
- [27] E. Khalil, H. Dai, Y. Zhang, B. Dilkina, and L. Song, "Learning combinatorial optimization algorithms over graphs," *Advances in neural information processing systems*, vol. 30, 2017.
- [28] G. Gordon and R. Tibshirani, "Karush-kuhn-tucker conditions," *Optimization*, vol. 10, no. 725/36, p. 725, 2012.
- [29] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.
- [30] F. Jiang, S. Tu, J. Zhang, L. Dong, K. Wang, K. Yang, and C. Pan, "M4sc: An mllm-based multi-modal, multi-task and multi-user semantic

communication system,” *IEEE Wireless Communications*, vol. 32, no. 5, pp. 40–47, 2025.

- [31] F. Jiang, C. Pan, K. Wang, P. Michiardi, O. A. Dobre, and M. Debbah, “From large ai models to agentic ai: A tutorial on future intelligent communications,” *IEEE Journal on Selected Areas in Communications*, vol. 44, pp. 3507–3540, 2026.
- [32] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, “Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter,” *arXiv preprint arXiv:1910.01108*, 2019.
- [33] W. Kool, H. van Hoof, and M. Welling, “Attention, learn to solve routing problems!” in *International Conference on Learning Representations*, 2018.
- [34] Y. Jin, Y. Ding, X. Pan, K. He, L. Zhao, T. Qin, L. Song, and J. Bian, “Pointerformer: Deep reinforced multi-pointer transformer for the traveling salesman problem,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 7, 2023, pp. 8132–8140.
- [35] F. Jiang, Y. Peng, L. Dong, K. Wang, K. Yang, C. Pan, D. Niyato, and O. A. Dobre, “Large language model enhanced multi-agent systems for 6g communications,” *IEEE Wireless Communications*, vol. 31, no. 6, pp. 48–55, 2024.



bile edge computing.

Feibo Jiang received his B.S. and M.S. degrees in School of Physics and Electronics from Hunan Normal University, China, in 2004 and 2007, respectively. He received his Ph.D. degree in School of Geosciences and Info-physics from the Central South University, China, in 2014. He is currently an associate professor at the Hunan Provincial Key Laboratory of Intelligent Computing and Language Information Processing, Hunan Normal University, China. His research interests include machine learning, Agentic AI, semantic communication, and mo-



Li Dong received the B.S. and M.S. degrees in School of Physics and Electronics from Hunan Normal University, China, in 2004 and 2007, respectively. She received her Ph.D. degree in School of Geosciences and Info-physics from the Central South University, China, in 2018. She is currently a professor at Hunan University of Technology and Business, China. Her research interests include machine learning, Internet of Things, and mobile edge computing.



Lei Mao is currently pursuing the M.S. degree with the School of Computer Science and Technology, Hunan Normal University. His main research interests include semantic communication and Agentic AI-enabled wireless communications.



Kezhi Wang received the Ph.D. degree from the University of Warwick, U.K., funded by the Chancellor’s International Scholarship. He is currently a Professor with the Department of Computer Science, Brunel University London, U.K. He is also a Royal Society Industry Fellow and has been recognised as a Highly Cited Researcher by Clarivate Web of Science and a Top 2% Scientist of the World. He has published over 150 papers in IEEE journals and has received several prestigious awards, including the IEEE Communications Society Leonard G. Abraham Prize, Heinrich Hertz Award, and Fred W. Ellersick Prize. His research interests include semantic communications, machine learning for communication systems, mobile edge computing, and wireless networks. He serves as an Associate Editor for IEEE Transactions on Vehicular Technology, IEEE Wireless Communications Letters, and IEEE Communications Letters, and has been actively involved in numerous IEEE conferences and editorial roles.



Xianbin Wang received his Ph.D. degree in electrical and computer engineering from the National University of Singapore in 2001. He has been with Western University, Canada, since 2008, where he is currently a Distinguished University Professor and Tier-1 Canada Research Chair in Trusted Communications and Computing. Prior to joining Western University, he was with the Communications Research Centre Canada as a Research Scientist and later a Senior Research Scientist from 2002 to 2007. From 2001 to 2002, he was a System Designer at STMicroelectronics. His current research interests include 5G/6G technologies, Internet of Things, machine learning, communications security, digital twin, and intelligent communications. He has over 700 highly cited journals and conference papers, in addition to over 30 granted and pending patents and several standard contributions.

Dr. Wang is a Fellow of the Canadian Academy of Engineering and a Fellow of the Engineering Institute of Canada. He has received many prestigious awards and recognitions, including the IEEE Canada R. A. Fessenden Award, Canada Research Chair, Engineering Research Excellence Award at Western University, Canadian Federal Government Public Service Award, Ontario Early Researcher Award, and twelve Best Paper Awards. He is currently a member of the Senate, Senate Committee on Academic Policy and Senate Committee on University Planning at Western. He also serves on NSERC Discovery Grant Review Panel for Computer Science. He has been involved in many flagship conferences, including IEEE GLOBECOM, ICC, VTC, PIMRC, WCNC, CCECE, and ICNC, in different roles, such as General Chair, TPC Chair, Symposium Chair, Tutorial Instructor, Track Chair, Session Chair, and Keynote Speaker. He was nominated as an IEEE Distinguished Lecturer multiple times by different societies including BTS, ComSoc and VTS. He serves/has served as the Editor-in-Chief, Associate Editor-in-Chief, Area Editor, and editor/associate editor for over ten journals. He was the Chair of the IEEE ComSoc Signal Processing and Computing for Communications (SPCC) Technical Committee and the Central Area Chair of IEEE Canada.



Abbas Jamalipour received the Ph.D. degree in electrical engineering from Nagoya University, Nagoya, Japan, in 1996. He holds the position of Professor of Ubiquitous Mobile Networking with The University of Sydney. He has authored nine technical books, eleven book chapters, over 550 technical papers, and five patents, all in the area of wireless communications and networking. He is a recipient of a number of prestigious awards, such as the 2019 IEEE ComSoc Distinguished Technical Achievement Award in Green Communications, the 2016 IEEE ComSoc Distinguished Technical Achievement Award in Communications Switching and Routing, the 2010 IEEE ComSoc Harold Sobol Award, the 2006 IEEE ComSoc Best Tutorial Paper Award, as well as over 15 Best Paper Awards. He was the President of IEEE Vehicular Technology Society from 2020 to 2021. Previously, he held the positions of Executive Vice-President and Editor-in-Chief of VTS Mobile World and has been an elected member of the Board of Governors of the IEEE Vehicular Technology Society since 2014. He was an Editor-in-Chief of IEEE Wireless Communications, the Vice President-Conferences, and a member of the Board of Governors of the IEEE Communications Society. He sits on the Editorial Board of IEEE Access and several other journals and is a member of the Advisory Board of IEEE Internet of Things Journal. He has been the General Chair or Technical Program Chair for several prestigious conferences, including IEEE ICC, GLOBECOM, WCNC, and PIMRC. Since January 2022, he has served as an Editor-in-Chief of the IEEE Transactions on Vehicular Technology. He is a Fellow of the Institute of Electrical, Information and Communication Engineers, a Fellow of Engineers Australia, an ACM Professional Member, and a Fellow of the International Artificial Intelligence Industry Alliance.