Optimizing UAV Data Collection through Redundancy-Aware Clustering and Scale-Adaptive Trajectory Optimization

Pengfei Wu[†], Jinyi Wang[†], Hong Zhu[†], Haiping Huang[†*], Chao Sha[†*]
[†]Nanjing University of Posts and Telecommunications, Nanjing, China
Email:{wupf,shac,hhp}@njupt.edu.cn

Abstract

Smart cities deploy numerous Internet of Things (IoT) devices that generate massive amounts of data requiring efficient collection methods, while Unmanned Aerial Vehicles (UAVs) face strict energy constraints and must navigate redundant information from overlapping sensor coverage areas. Our work tackles a fundamental tension between maximizing effective data collection and minimizing UAV energy consumption in environments where IoT sensor coverage areas overlap, creating redundant data. We propose a novel two-stage optimization framework that balances this trade-off through complementary algorithmic components. Our submodular maximization-based approach selects optimal waypoints by grouping IoT devices according to data similarity patterns, minimizing redundant transmissions while maximizing collection coverage. The framework integrates this with an adaptive trajectory optimization technique combining Graph Convolutional Networks and Monte Carlo Tree Search, enabling efficient flight path planning without requiring retraining when network scales change. Comprehensive simulations demonstrate that our approach significantly outperforms baseline methods in reducing UAV energy consumption while increasing effective data collection, maintaining robust performance across varying network scales.

CCS Concepts

• **Networks** → **Wireless access networks**; • **Computer systems organization** → *Sensor networks*; • **Theory of computation** → Scheduling algorithms.

Keywords

Unmanned Aerial Vehicles, Data Collection, Trajectory Optimization, Smart Cities, Internet of Things

ACM Reference Format:
Pengfei Wu[†], Jinyi Wang[†], Hong Zhu[†], Haiping Huang[†*], Chao Sha[†*]. 2025. Optimizing UAV Data Collection through Redundancy-Aware Clustering and Scale-Adaptive Trajectory Optimization. In *Proceedings of Energy-Efficient UAV Data Collection: A Heatmap-Based Trajectory Optimization Approach (MOBIHOC '25).* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

MOBIHOC '25, Houston, USA

ACM ISBN 978-x-xxxx-xxxx-x/YYYY/MM
https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 Introduction

The widespread application of IoT devices in modern cities has accelerated the transformation of traditional urban areas into smart cities [1]. These technology-enhanced environments aim to improve urban management efficiency, resident quality of life, and resource allocation through integrated IoT systems [2]. The key to this transformation lies in the efficient collection and analysis of the vast amounts of data generated by IoT sensors deployed throughout the city. However, due to their portable design, most IoT devices face limitations in energy, computing power, and storage capacity. In addition, high energy consumption during data transmission makes multi-hop or relay-based data forwarding to base stations impractical [3]. Therefore, how to efficiently utilize the data from these devices has become a major challenge in the development of smart cities.

Recently, UAVs have emerged as a promising approach to address this challenge. Compared to traditional data collection methods, UAV-assisted data collection offers several significant advantages [4–7]. UAVs enable efficient and flexible data collection through optimized flight trajectories that significantly reduce data collection time. The aerial platforms also help minimize the energy consumption associated with IoT data transmission, thus extending the operational life of IoT networks. Additionally, due to their ability to operate at high altitudes, UAVs have a higher probability of establishing line-of-sight (LoS) connections with ground-based sensors, improving communication reliability and expanding coverage. These advantages position UAVs as an optimal solution for smart city data collection. Despite the significant potential of UAVs in IoT data collection, several challenges remain. UAVs rely on battery power, which imposes strict energy constraints and limits practical data collection time when consumption is excessive [8, 9]. The overlapping coverage of sensors also generates redundant data, creating inefficiencies in the collection process [10]. Balancing effective data collection with energy consumption thus emerges as a critical challenge requiring innovative solutions.

Existing studies approach this problem through a two-stage UAV data collection process [11–13]. Initially, clustering algorithms divide IoT devices into distinct groups with designated Cluster Heads (CHs), establishing optimal UAV hovering positions. The process then shifts to calculating the shortest flight path connecting these waypoints. The UAV subsequently traverses this optimized route, visiting each cluster sequentially to gather data [14]. However, these approaches overlook redundant data collection while lacking precision and generalization in trajectory optimization, particularly at larger scales. Current clustering methods prioritize geographical distance over data correlation among neighboring sensors, leading to unnecessary data redundancy. Traditional trajectory optimization algorithms struggle with computational efficiency and solution quality in large-scale networks. The requirement for complete model retraining when network size changes further limits their practical utility in dynamic urban environments. As smart cities evolve, IoT networks grow increasingly dense and complex, intensifying challenges in data redundancy management and energy efficiency. Future UAV-assisted data collection systems must address current limitations while adapting to the expanding complexity of urban IoT deployments. This necessitates sophisticated approaches capable of automatic scaling across varying network sizes while maintaining optimal efficiency in both data collection and energy utilization.

To address these challenges, this paper investigates UAV-assisted IoT data collection, focusing on the critical trade-off between collection efficiency and energy consumption. We propose a novel strategy that integrates data redundancy management with path optimization to enhance overall system performance. Our approach first formulates the optimization problem and methodically decomposes it into two distinct sub-problems. We then develop specialized algorithms for each sub-problem and validate our solution through comprehensive simulation experiments, demonstrating significant improvements in both data collection efficiency and energy utilization. The contributions of our work can be summarized as:

- We propose a two-stage optimization framework for UAV-assisted IoT networks that balances data collection efficiency with energy consumption while effectively addressing data redundancy issues.
- We develop MISRED (Maximizing the Suppression of Network Redundancy), a submodular maximization-based algorithm for optimal waypoint selection that minimizes redundant transmissions while maximizing collection coverage.
- We introduce a Heatmap-Guided Reinforcement Learning trajectory optimization approach, UTICA, combining Graph Convolutional Networks with Monte Carlo Tree Search, enabling adaptive flight trajectory optimization across varying network scales without retraining requirements.
- We validate our approach through comprehensive simulations, demonstrating significant performance improvements over baseline methods in redundancy reduction, scalability, generalization, and autonomous operation.

2 PRELIMINARIES

2.1 System Model

We consider a UAV-assisted IoT network consisting of a UAV and n IoT devices, applied to scenarios such as smart cities or environmental monitoring. The set of IoT devices is denoted as $V = \{v_1, v_2, \dots, v_n\}$, where the coordinates of IoT device v_i are represented as $(x_i, y_i, 0)$, for $1 \le i \le n$.

To optimize UAV energy consumption and minimize redundant data collection, we partition the n IoT devices into K clusters, each comprising one main IoT device and multiple subordinate devices. The UAV will collect data at a height H above the main IoT device of each cluster. Consequently, we define a set of waypoints $C = \{c_1, c_2, \dots, c_K\}$, where the coordinates of waypoint c_k are $(x_k^{'}, y_k^{'}, H)$, corresponding to the K clusters $Q_1, Q_2, \dots, Q_K$.

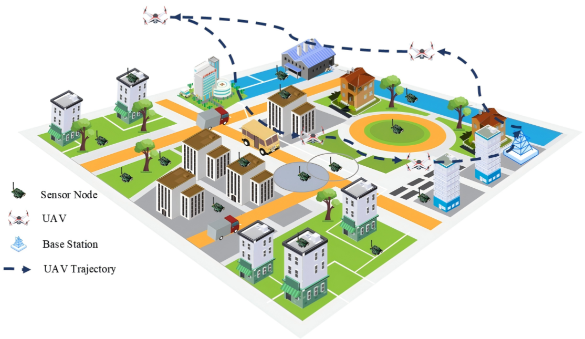


Figure 1: The UAV-assisted IoT data collection system in a smart city illustrates the distribution of IoT devices with overlapping coverage areas due to their geographical proximity.

Within this network, the UAV is scheduled to follow predetermined trajectory, gathering data generated by IoT devices. Departing from the starting point, the UAV flies at a constant velocity v to a designated waypoint where it conducts data collection. Due to the UAV's limited energy, it is unable to collect data from all IoT devices. Instead, it selectively gathers non-redundant data. Upon completing its mission, the UAV returns to the starting point for energy replenishment.

For any given cluster Q_i, the UAV can collect data from both the primary and secondary sensors. When the UAV hovers above cluster Q_i, the primary sensor within that cluster first transmits its data to the UAV. Simultaneously, the secondary IoT devices listen to the primary sensor's transmission, enabling them to identify which data is redundant. Consequently, when transmitting data to the UAV, the secondary sensors only need to send their non-redundant portions. This approach effectively reduces data redundancy and enhances energy efficiency.

2.2 Spatial Data Correlation Model

In wireless sensor networks, geographically adjacent sensors may have similar sensor data. According to the study in [10], the spatial data correlation $c(i, j)$ is calculated by evaluating the data similarity between sensor v_i and sensor v_j over the same time period. This can be expressed by the following formula:

$$c(i, j) = \frac{N_{i,j}}{N}, \tag{1}$$

where N represents the number of time slots within a fixed time period, and $N_{i,j}$ denotes the number of time slots in which the data difference between sensor v_i and sensor v_j does not exceed the threshold γ over the N time slots.

To collect non-redundant information from the n sensors in the network, we partition the n IoT devices in the network into K disjoint clusters $Q_1, Q_2, \dots, Q_K$. Each cluster contains a main sensor and several subordinate sensors. When the UAV collects data in a cluster, the main sensor transmits all data to the UAV, while subordinate sensors only transmit their non-redundant data.

This reduces the amount of redundant data collected from each cluster, thereby lowering the energy consumption of the UAV and enabling it to collect more data from the IoT devices.

We apply the method from [10] to partition the n IoT devices in the network into clusters. The number of clusters K is not predetermined but is determined by the data correlation relationship. Specifically, let u_i be the main sensor of cluster Q_i. The redundant data between u_i and subordinate sensors v_j in the same cluster can be expressed as:

$$r_{ij} = \min\{D_i^Q, D_j^Q\} \cdot c(i,j). \tag{2}$$

It is important to note that our calculation method differs from [10], where if the data collected by the main sensor is significantly larger than that collected by subordinate sensors, the calculated redundant data between them may exceed the total data collected by the subordinate sensors. Our approach, while potentially underestimating the actual redundancy, effectively avoids this issue. Therefore, after determining the main sensor and subordinate IoT devices, the total amount of redundant data transmission in the network is given by:

$$\sum_{i=1}^{K} \sum_{v_j \in Q_i \setminus \{u_i\}} r_{ij}. \tag{3}$$

2.3 UAV Model

Without loss of generality, we assume that the UAV flies from one waypoint to another at a constant speed v_{uav}. The propulsion energy consumption of the UAV is given by the [15]:

$$P_{\text{mov}}(v_{\text{uav}}) = P_0 \left(1 + \frac{3v_{\text{uav}}^2}{U_{\text{tip}}^2}\right) + P_1 \left(\left(1 + \frac{v_{\text{uav}}^4}{4v_0^4}\right)^{1/2} - \frac{v_{\text{uav}}^2}{2v_0^2}\right)^{1/2} + \frac{1}{2} d_0 \rho s \delta v_{\text{uav}}^3 , \tag{4}$$

where P_0 and P_1 are predefined constants representing the profile power and induced power of the rotor blades in hover, respectively; U_{tip} denotes the blade tip speed; v_0 is the known average rotor-induced velocity in hover; d_0 and s represent the body drag coefficient and rotor stiffness, respectively; ρ is the air density, and δ refers to the rotor disk area.

When the UAV collects data at the waypoint c_i located above cluster Q_i, the power consumption during hovering can be obtained by substituting $v_{uav} = 0$ into the equation. Furthermore, if the UAV's hovering altitude H is relatively low, the variation in data transmission rates between the main IoT device and its subordinate devices within the cluster can be considered negligible. To simplify the discussion, we assume that all IoT devices in a cluster have identical transmission rates. Additionally, the UAV's hovering time is assumed to be equal to the data transmission time. Accordingly, the energy consumption during data collection can be expressed as:

$$E_{c_k} = \frac{D_i^{nr}}{r_{\text{data}}} (P_{\text{hover}} + P_{\text{com}}), \tag{5}$$

where D_i^{nr} represents the amount of non-redundant data collected by the UAV at the waypoint c_i, and P_{com} denotes the communication power of the UAV. The energy consumption of the UAV when moving from waypoint c_i to another waypoint c_j can be represented as:

$$E_{c_i,c_j} = \frac{\|c_i - c_j\|}{v_{\text{uav}}} P_{\text{move}}, \tag{6}$$

where $\|c_i - c_j\|$ denotes the Euclidean distance between waypoints c_i and c_j. Let $C = \{c_1, c_2, \ldots, c_K\}$ denote the collection of waypoints, and L_{c_i,c_j} indicate whether the UAV travels from c_i to c_j. The specific definition of L_{c_i,c_j} is given as:

$$L_{c_i,c_j} = \begin{cases} 1, & \text{if the path goes from } c_i \text{ to } c_j. \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

2.4 Communication Model

Before collecting data from the IoT devices, the UAV must first establish a reliable connection with each IoT device. In this paper, we focus on data collection by the UAV during hovering. Two types of communication links are considered: Line-of-Sight (LoS) and Non-Line-of-Sight (NLoS). When the UAV is flying at a higher altitude, fewer obstacles are present, and an LoS link is used. Otherwise, an NLoS link is employed. The probability of an LoS link is typically given by the following formula:

$$P_{\text{LoS}} = \frac{1}{1 + a \exp\left(-b\left[\frac{180}{\pi} \arcsin\left(\frac{H}{d_{i,k}}\right) - a\right]\right)}, \tag{8}$$

where a and b are two environment-dependent parameters, and $d_{i,k}$ is the distance between the UAV's waypoint and the IoT devices. Meanwhile, the probability of an NLoS link is given by $P_{\text{NLoS}} = 1 - P_{\text{LoS}}$. The average path loss between each IoT device and the UAV can be expressed as:

$$\overline{P}_{\text{loss}} = P_{\text{LoS}} \left(K_0 + \mu_{\text{LoS}}\right) + P_{\text{NLoS}} \left(K_0 + \mu_{\text{NLoS}}\right), \tag{9}$$

where μ_{LoS} and μ_{NLoS} represent the average excess path loss for LoS and NLoS links, respectively, and $K_0 = 10\alpha \log_{10}(\frac{4\pi f_c H}{c})$, where α is the path loss exponent, c is the speed of light, and f_c is the carrier frequency. Therefore, the average data transmission rate between each IoT device and the UAV can be calculated using the following formula:

$$r_{\text{data}} = B_{\text{width}} \log_2 \left(1 + \frac{P_{trans}}{\overline{P}_{\text{loss}} N_0}\right), \tag{10}$$

where B_{width} is the available bandwidth, N_0 is the noise power spectral density, and P_{trans} is the transmission power of IoT devices.

3 Problem Formulation

In this paper, our objective is to collect data from all IoT devices using a UAV while minimizing redundant data collection and energy consumption within a reasonable time. The optimization goal can be formulated as:

$$\mathcal{P}_1 : \min_{\mathcal{U},\Pi,\text{S}} \left[\sum_{i=1}^{K} R_{\text{collect}}^i(\text{S}) + E_{\text{hover}}(\text{S}) + E_{\text{flight}}(\Pi, \mathcal{U})\right], \tag{11}$$

where $\mathcal{U}$ represents the set of waypoints selected by the UAV during data collection, Π denotes the visiting sequence of these waypoints, and S characterizes the data similarity between different IoT devices. By optimizing the above equation, we aim to minimize both redundant data collection and energy consumption. The amount of redundant data collected, $\sum_{i=1}^{K} R^i_{\text{collect}}(\mathrm{S})$, and the hovering energy consumption, $E_{\text{hover}}(\mathrm{S})$, are both determined by the data similarity matrix S. Specifically, the hovering energy consumption is primarily influenced by the amount of data collected at each waypoint, which is closely related to S. Moreover, the energy consumption incurred during data transmission is relatively small. Therefore, in the optimization objective, the minimization of these two energy-related components mainly depends on S and is independent of the UAV's flight trajectory Π and the set of waypoints $\mathcal{U}$. Thus, the original optimization problem can be decomposed as:

$$\begin{aligned}
&\min_{\mathcal{U},\Pi,\mathrm{S}} \left[\sum_{i=1}^{K} R^i_{\text{collect}}(\mathrm{S}) + E_{\text{hover}}(\mathrm{S}) + E_{\text{flight}}(\Pi,\mathcal{U})\right] \\
&= \min_{\mathrm{S}} \left[\sum_{i=1}^{K} R^i_{\text{collect}}(\mathrm{S}) + E_{\text{hover}}(\mathrm{S})\right] + \min_{\mathcal{U},\Pi} E_{\text{flight}}(\Pi,\mathcal{U})
\end{aligned} \tag{12}$$

By substituting equations (4), (5), (6), and (7) into the above formulation, we obtain:

$$\begin{aligned}
&\min_{\mathcal{U},\Pi,\mathrm{S}} \left[\sum_{i=1}^{K} R^i_{\text{collect}}(\mathrm{S}) + E_{\text{hover}}(\mathrm{S}) + E_{\text{flight}}(\Pi,\mathcal{U})\right] \\
&= \min \sum_{i=0}^{K}\sum_{j=0}^{K} L_{c_i,c_j} \|c_i - c_j\| \left[\frac{P_0}{v_{\text{uav}}} + \frac{3P_0 v_{\text{uav}}}{U_{\text{tip}}^2}\right. \\
&\quad \left. + \frac{P_1}{v_{\text{uav}}}\left(\sqrt{1+\frac{v_{\text{uav}}^4}{4v_0^4}} - \frac{v_{\text{uav}}^2}{2v_0^2}\right)^{\frac{1}{2}} + \frac{d_0 \rho s \delta v_{\text{uav}}^2}{2}\right] \\
&\quad + \min\left[\sum_{i=1}^{K} \frac{D_i}{r_{\text{data}}}(P_0 + P_1 + P_{\text{com}}) + R^i_{\text{collect}}\right]
\end{aligned} \tag{13}$$

To simplify the notation, we define $f(v_{uav})$ as follows:

$$\begin{aligned}
f(v_{\text{uav}}) =& \frac{P_0}{v_{\text{uav}}} + \frac{3P_0 v_{\text{uav}}}{U_{\text{tip}}^2} \\
&+ \frac{P_1}{v_{\text{uav}}}\left(\left(1+\frac{v_{\text{uav}}^4}{4v_0^4}\right)^{\frac{1}{2}} - \frac{v_{\text{uav}}^2}{2v_0^2}\right)^{\frac{1}{2}} \\
&+ \frac{d_0 \rho s \delta v_{\text{uav}}^2}{2}
\end{aligned} \tag{14}$$

Thus, the optimization problem can be rewritten as:

$$\begin{aligned}
&\min_{\mathcal{U},\Pi,\mathrm{S}} \left[\sum_{i=1}^{K} R^i_{\text{collect}}(\mathrm{S}) + E_{\text{hover}}(\mathrm{S}) + E_{\text{flight}}(\Pi,\mathcal{U})\right] \\
&= \min \sum_{i=0}^{K}\sum_{j=0}^{K} L_{c_i,c_j} \|c_i - c_j\| f(v_{\text{uav}}) \\
&\quad + \min\left[\sum_{i=1}^{K} \frac{D_i}{r_{\text{data}}}(P_0 + P_1 + P_{\text{com}}) + R^i_{\text{collect}}\right]
\end{aligned} \tag{15}$$

From the above formula, it is evident that minimizing redundant data collection and energy consumption is closely related to both the flight distance of the UAV and the amount of redundant data suppressed from transmission within the network. The greater the suppressed redundant data, the less redundant data is collected. Given a fixed total data volume, an increase in the suppressed redundant data leads to a corresponding reduction in the collected data volume. To address this issue, we decompose it into two subproblems. First, clustering is conducted based on the data correlation among IoT devices to determine the set of UAV waypoints $\mathcal{U}$, thereby aiming to minimize redundant data transmission within the network. Secondly, based on the clustering results, the waypoints of the UAV are scheduled to optimize its flight trajectory, denoted as Π. In the next section, we will elaborate on the specific methods for addressing these two subproblems.

4 SOLUTION METHOD

4.1 Maximizing the Suppression of Network Redundancy

In this section, we propose an algorithm named MISRED[1] for optimizing the selection of waypoints for UAVs, aiming to maximize the suppression of redundant data transmission in the network. The primary objective of this algorithm is to partition all IoT devices into K dedicated clusters, denoted as $Q_1, Q_2, \ldots, Q_K$, and select a representative IoT device within each cluster Q_i to minimize redundant data transmissions. This approach effectively reduces the energy consumption of the UAV during the hovering-based data collection process. Subsequently, the UAV's K waypoints are determined based on the locations of these representative IoT devices to optimize data collection efficiency. Mathematically, this optimization process can be formulated as:

$$\mathcal{P}_2: \quad \max_{M\subseteq V}\{f(M)\} = \max_{M\subseteq V}\left\{\sum_{v_j\in V\setminus M} \max_{v_i\in M}\{r_{ij}\}\right\}, \tag{16}$$

subject to the communication range constraint that ensures connectivity between IoT devices within each cluster.

The problem $\mathcal{P}_2$ is an unconstrained submodular maximization problem, which has been proven to be NP-hard [10]. According to this research, maximizing redundant data suppression in the network is equivalent to selecting a set of primary sensors $M \subseteq V$ such that $\mathcal{P}_2$ holds. Leveraging this equivalence, we propose MISRED to optimize the selection of UAV waypoints. Algorithm 1 details the MISRED approach. The algorithm takes as input the set of IoT devices, the amount of suppressed redundant data transmissions when assigning master-slave relationships, an error ratio ϵ, and a desired success probability. MISRED operates through an iterative probabilistic process over L iterations. In each iteration, the algorithm evaluates each device's potential contribution to the objective function by calculating its marginal gain when added to the master set and its marginal loss when removed from consideration. Based on these values, it makes probabilistic decisions to either include devices in the master set or exclude them. For each non-master device, MISRED assigns a master device that maximizes redundant data suppression. Finally, it determines the optimal UAV waypoints directly above the selected master devices at a predetermined height H.

[1] The name MISRED comes from MaxImizing the Suppression of Network REDundancy.

Algorithm 1 MISRED: Maximizing the Suppression of Network Redundancy

```
Input: IoT devices $v_1, v_2, \ldots, v_n$, the amount $p_{ij}$ of suppressed redundant data transmissions by sensor $v_j$ if $v_i$ is its master IoT device, an error ratio $\epsilon$ with $0 < \epsilon \le 0.5$, and a success probability $1 - \frac{1}{n^\alpha}$.
Output: Master IoT device set $M$ and waypoint set $C$.
Let $L = \epsilon \cdot \log_{1+2\epsilon} n$;
Let $M = \emptyset$ and $N_0 = V$;
for $l \leftarrow 1$ to $L$ do
    Let $M_0 = M$ and $N_0 = V \setminus M$;
    for $i \leftarrow 1$ to $n$ do
        $a_i = f(M_{i-1} \cup \{v_i\}) - f(M_{i-1})$; $\tilde{a}_i = \max\{a_i, 0\}$;
        $b_i = f(N_{i-1} \setminus \{v_i\}) - f(N_{i-1})$; $\tilde{b_i} = \max\{b_i, 0\}$;
        Generate a random number $r \in [0, 1]$
        if $r \le \frac{\tilde{a}_i}{\tilde{a}_i + \tilde{b}_i}$ then
            $M_i = M_{i-1} \cup \{v_i\}$
        else
            $N_i = N_{i-1} \setminus \{v_i\}$
        end if
    end for
    $M^{(l)} = M_n$;
    for each $v_j \in V \setminus M^{(l)}$ do
        $u = \arg\max_{v_i \in M^{(l)}} \{p_{ij}\}$;
    end for
    if $f(M^{(l)}) > f(M)$ then
        $M \leftarrow M^{(l)}$;
    end if
end for
$C = \{(x_i, y_i, H) \mid v_i \in M\}$;
return the sets of master IoT devices $M$ and waypoints $C$;
```

The MISRED algorithm determines the hovering locations by maximizing redundant data suppression and guarantees a $(0.5-\epsilon)$-approximate solution with at least $1-(1/n^\alpha)$ probability, where ϵ and α are given constants satisfying $0 < \epsilon \le 0.5$ and $\alpha > 0$. In terms of computational complexity, MISRED executes $L = \epsilon \cdot \log_{1+2\epsilon} n$ iterations, with each iteration requiring $O(n^2)$ operations for calculating marginal gains and losses across all IoT devices. Thus, the overall time complexity of MISRED is $O(n^2 \cdot \log n)$, making it efficient for practical deployment in medium to large-scale IoT networks.

4.2 Heatmap-Guided Reinforcement Learning Trajectory Optimization

After determining the UAV's hover positions by solving problem $\mathcal{P}_2$, we propose a trajectory optimization algorithm, termed UTICA[2], to optimize the UAV's flight energy consumption. The UAV begins at a designated starting point, traverses all waypoints to collect data, and returns to the starting point upon task completion. Let $C' = \{c_0, c_1, \ldots, c_K\}$ denote the set of all hover positions including the starting point c_0. The optimization objective for flight energy consumption is expressed as follows:

$$\mathcal{P}_3: \quad \min E_{\text{flight}} = \sum_{i=0}^{K} \sum_{j=0}^{K} E_{c_i,c_j} L_{c_i,c_j} \tag{17}$$

subject to the following constraints:

$$\sum_{i=0, i\neq j}^{K} L_{c_i,c_j} = 1, \quad \forall c_j \in C'; \tag{17a}$$

$$\sum_{j=0, j\neq i}^{K} L_{c_i,c_j} = 1, \quad \forall c_i \in C'; \tag{17b}$$

$$\sum_{c_i \in S} \sum_{c_j \in S} L_{c_i,c_j} \le |S| - 1, \quad \forall \boldsymbol{S} \subset \boldsymbol{C'}, |\boldsymbol{S}| \ge 2; \tag{17c}$$

$$L_{c_i,c_j} \in \{0, 1\}; \tag{17d}$$

$$E_{\text{flight}} + E_{\text{hover}} \le E_{\text{max}}. \tag{17e}$$

Constraints (17a) and (17b) ensure that each hover location is visited exactly once and departed exactly once. Constraint (17c) prevents subloops (or subtours), guaranteeing the connectivity and validity of the path. Constraint (17d) restricts the variable L_{c_i,c_j} to binary values, and Constraint (17e) ensures that the total energy consumption, including both flight and hovering energy, does not exceed the maximum allowable energy.

Problem $\mathcal{P}_3$ represents a classic combinatorial optimization challenge, the Traveling Salesman Problem (TSP), which is known to be NP-hard. In our UAV context, the objective is to determine the most energy-efficient flight path that visits each waypoint exactly once while satisfying the energy constraints. To address this computational challenge, our proposed UTICA algorithm leverages a heatmap-guided reinforcement learning approach that efficiently navigates the solution space without exhaustive path enumeration. This approach is particularly well-suited for resource-constrained UAV systems operating in IoT networks where energy efficiency is critical.

The UTICA algorithm operates through two distinct phases. The first phase involves heatmap generation, which employs a Graph Convolutional Network with an attention mechanism to compute the probability heatmap for the set of waypoints. The heatmap is defined as an $n \times n$ matrix P, where each element $P_{ij} \in [0, 1]$ represents the probability that edge (i, j) belongs to the optimal path solution. The second phase focuses on traversal order optimization, which implements a Monte Carlo Tree Search augmented with policy gradients to determine the optimal visiting sequence of waypoints based on the heatmap generated in the first phase.

4.2.1 Heatmap generation. Given a set of UAV waypoints represented as a graph G in a two-dimensional plane with coordinates defined as C', we generate the probability heatmap for graph G using the heatmap generator, as shown in Figure 2.

To achieve this, we trained a GCN model with an integrated attention mechanism on a large dataset of TSP instances with known optimal solutions, specifically tailored for fixed-scale problem heatmap generation. To accommodate varying problem scales, we implemented graph sampling, transformation, and heatmap merging techniques to produce a probabilistic heatmap aligned

[2] The name UTICA comes from heatmap-gUided reinforcemenT learnIng trajeCtory optimizAtion

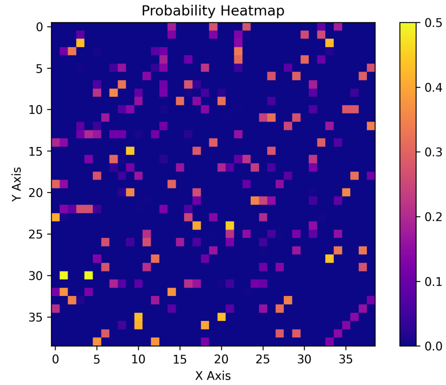


Figure 2: The probability heatmap depicts the likelihood of edges between waypoints belonging to the optimal UAV trajectory. Yellow coloration indicates higher probability of inclusion in the optimal solution.

with the given waypoints. The proposed GCN model for generating heatmaps to address the fixed-size m problem is described as follows:

Input Layer. The input layer processes node and edge features from the graph. Each node m_i is represented by its 2D coordinates $m_i \in [0,1]^2$, linearly mapped to a higher-dimensional feature space $\mathbb{R}^h$ through:

$$\alpha_i = A_1 m_i + b_1, \tag{18}$$

where $A_1 \in \mathbb{R}^{h\times 2}$ is a learnable weight matrix and $b_1 \in \mathbb{R}^h$ is a bias vector. To normalize node features and mitigate internal covariate shift, we apply Batch Normalization:

$$\alpha_i' = \mathrm{BN}(\alpha_i). \tag{19}$$

For each edge (m_i, m_j), the edge feature β_{ij} consists of two components: the Euclidean distance d_{ij} between the nodes and the k-nearest neighbor relationship between the node pair. The edge feature is calculated as:

$$\beta_{ij} = A_2 d_{ij} + b_2 \parallel A_3 \delta_{ij}^{\mathrm{k-NN}}, \tag{20}$$

where $A_2 \in \mathbb{R}^{\frac{h}{2}\times 1}, A_3 \in \mathbb{R}^{\frac{h}{2}\times 3}$. The distance d_{ij} embeds into a $\frac{h}{2}$-dimensional feature vector, while $\delta_{ij}^{\mathrm{k-NN}}$ encodes the k-nearest neighbor relationship between nodes m_i and m_j: $\delta_{ij}^{\mathrm{k-NN}} = 1$ for k-nearest neighbors, $\delta_{ij}^{\mathrm{k-NN}} = 2$ for self-loops, and $\delta_{ij}^{\mathrm{k-NN}} = 0$ otherwise. The symbol $\parallel$ denotes concatenation. To maintain consistent feature scaling, we apply batch normalization:

$$\beta_{ij}' = \mathrm{BN}(\beta_{ij}). \tag{21}$$

Convolutional Layer. In each GCN layer, node and edge features are updated based on adjacency relationships. Let m_i^ℓ denote the node feature at layer ℓ, and e_{ij}^ℓ the edge feature. The node features are updated as:

$$m_i^{\ell+1} = m_i^\ell + \mathrm{ReLU}\Big(\mathrm{BN}\Big(W_1^\ell m_i^\ell + \textstyle\sum_{i\sim j} \eta_{ij}^\ell \odot W_2^\ell m_j^\ell\Big)\Big) \ , \tag{22}$$

where the attention weight η_{ij}^ℓ is defined as:

$$\eta_{ij}^\ell = \frac{\sigma(e_{ij}^\ell)}{\sum_{j'\sim i} \sigma(e_{ij'}^\ell) + \varepsilon}. \tag{23}$$

This weight controls neighboring node m_j's influence on node m_i, enabling adaptive aggregation. $W_1^\ell, W_2^\ell \in \mathbb{R}^{h\times h}$ are learnable weight matrices, and ReLU introduces non-linearity. Edge features are updated by:

$$e_{ij}^{\ell+1} = e_{ij}^\ell + \mathrm{ReLU}\Big(\mathrm{BN}\Big(W_3^\ell e_{ij}^\ell + W_4^\ell m_i^\ell + W_5^\ell m_j^\ell\Big)\Big). \tag{24}$$

Batch normalization ensures stability, while residual connections retain original feature information:

$$m_i^{\ell+1} = m_i^{\ell+1} + m_i^\ell, \tag{25}$$

$$e_{ij}^{\ell+1} = e_{ij}^{\ell+1} + e_{ij}^\ell. \tag{26}$$

MLP Layer. The final edge embedding e_{ij}^L computes the connection probability of edge (m_i, m_j) in the shortest path within the input graph. This probability forms the heatmap on the adjacency matrix of path connections. Each edge's shortest path probability $p_{ij}^{\mathrm{opt}} \in [0,1]$ is calculated via:

$$p_{ij}^{\mathrm{opt}} = \mathrm{MLP}(e_{ij}^L). \tag{27}$$

Loss Function. During training, each TSP instance provides a ground truth shortest path permutation π, converted to an adjacency matrix where $\hat{p}_{ij}^{\mathrm{opt}} = 1$ if an edge exists between nodes m_i and m_j, and $\hat{p}_{ij}^{\mathrm{opt}} = 0$ otherwise. We minimize the weighted binary cross-entropy loss, computed as the mean loss over mini-batches. As problem scale increases, negative samples grow significantly, causing class imbalance. To address this, we incorporate balanced class weights: for negative samples, $w_0 = n/(n-2) \times c$; for positive samples, $w_1 = n/2 \times c$, where $c = 2$ represents the total number of classes. These weights effectively balance the positive-negative sample ratio, enabling accurate learning of shortest path features.

Once the network model is trained, it generates heatmaps for m-node problems. To accommodate varying problem scales, we employ graph sampling, transformation, and sub-heatmap merging techniques.

Step 1: Graph Sampling. Multiple m-node subgraphs G' are extracted from the original graph G. During each iteration, the least-selected node c (with minimum selection count O_c) serves as the cluster center, and its $m-1$ nearest neighbors are selected using the k-NN algorithm. This process continues until all nodes reach a predefined selection threshold, ensuring balanced coverage of the original graph.

Step 2: Graph Transformation. Each subgraph G' is transformed into G'' with nodes mapped to a unit square. After calculating the bounding box coordinates $(x^{min}, x^{max}, y^{min}, y^{max})$, an amplification factor $s = 1/\max(x^{max} - x^{min}, y^{max} - y^{min})$ transforms each node from (x_i, y_i) to $(x_i^{new}, y_i^{new}) = (s \times (x_i - x^{min}), s \times (y_i - y^{min}))$. Each transformed graph is then processed by our trained model to generate a corresponding sub-heatmap.

Step 3: Sub-Heatmap Merging. By repeating the above steps, we obtain I sub-heatmaps collected in set L_h. The probability P_{ij} of edge (i, j) belonging to the shortest path is computed by:

$$P_{ij} = \frac{1}{O_{ij}} \times \sum_{l=1}^{I} P''_{ij}(l), \tag{28}$$

where $P''_{ij}(l)$ represents the optimal solution probability for edge (i, j) in the l-th sub-heatmap and O_{ij} tracks the number of times edge (i, j) appears across all subgraphs. Edges with $P_{ij} < 10^{-4}$ are removed to reduce the search space.

4.2.2 *Reinforcement Learning based Trajectory Optimization .*

To optimize the UAV's traversal sequence, we utilize a reinforcement learning-driven path search method that leverages the heatmap corresponding to the generated set of waypoints. This search process is modeled as a Markov Decision Process (MDP), where actions $\boldsymbol{a}$ are applied iteratively starting from an initial state $\boldsymbol{\pi}$, gradually optimizing the path until a target state $\boldsymbol{\pi}^*$ is reached. The relevant function designs are described as follows:

State Function. The state function defines a complete solution for the UAV traversal sequence across all waypoints, represented as a permutation of waypoints $\boldsymbol{\pi} = (\pi_1, \pi_2, \ldots, \pi_n)$. Each state represents a potential UAV path that starts at the initial point, traverses each waypoint exactly once, and returns to the starting point.

Action Function. The action function transforms one state $\boldsymbol{\pi}$ to another state $\boldsymbol{\pi}^*$. In UAV trajectory optimization, an action represents a $k-$opt operation $(2 \leq k \leq n)$ that replaces k edges with k new edges to form a different path. An action $\boldsymbol{a}$ is denoted as $\boldsymbol{a} = (a_1, b_1, a_2, b_2, \ldots, a_k, b_k, a_{k+1})$, where $a_{k+1} = a_1$ ensures the path remains closed.

While these k-opt operations provide a mechanism for path modification, efficiently navigating the vast solution space remains challenging, particularly as k increases. The key challenge lies in selecting which edges to replace among the exponential number of possibilities. To address this, we develop a learning-based approach that intelligently samples promising actions according to historical performance. Algorithm 2 presents an integrated framework that combines local neighborhood search with Monte Carlo Tree Search (MCTS) for efficient exploration of the solution space.

Algorithm 2 begins by initializing weight matrix W from heatmap probabilities and visit matrix Q to track edge selection frequency. An initial solution is generated by sequentially selecting nodes with probabilities proportional to $\exp(P_{\pi_{ij}})$. The optimization process then alternates between two phases: a simple 2-opt neighborhood search for local improvements, and an MCTS-based exploration when local optima are encountered.

In the MCTS phase, actions are constructed incrementally through probabilistic node selection. For each step, the algorithm computes a potential value $Z_{b_{ij}}$ that balances exploitation of high-weight edges with exploration of less-visited options:

$$Z_{b_{ij}} = \frac{W_{b_{ij}}}{\Omega_{b_i}} + \alpha \sqrt{\frac{\ln(M+1)}{Q_{b_{ij}} + 1}}, \tag{29}$$

where Ω_{b_i} represents the average edge weight from node b_i. Candidate nodes form set X where $W_{b_{ij}} \geq 1$, excluding already connected nodes. Selection probabilities are computed as:

$$P_j = \frac{Z_{b_{ij}}}{\sum_{l \in X} Z_{b_i l}}. \tag{30}$$

When an improving action is found, the algorithm updates edge weights proportionally to the improvement quality:

$$W_{b_i a_{i+1}} \leftarrow W_{b_i a_{i+1}} + \beta \left[\exp\left(\frac{L(\boldsymbol{\pi}) - L(\boldsymbol{\pi}^{new})}{L(\boldsymbol{\pi})} \right) - 1 \right]. \tag{31}$$

This adaptive learning mechanism progressively biases the search toward promising solution regions while maintaining sufficient exploration through the UCB term in the potential function. The visit matrix Q tracks edge selection frequency, ensuring balanced exploration of the search space over time.

Algorithm 2 UTICA: Heatmap-Guided Reinforcement Learning Trajectory Optimization

```
Input: Heatmap $P$, time limit $T$, parameters $\alpha$, $\beta$
Output: Optimal traversal path $\boldsymbol{\pi}^*$
Initialize: $W_{ij} = 100 \times P_{ij}$, $Q_{ij} = 0$ for all $i, j$, $M = 0$
Generate initial path $\boldsymbol{\pi}$ with probability $\propto \exp(P_{\pi_{ij}})$
$\boldsymbol{\pi}^* \leftarrow \boldsymbol{\pi}$
while time $< T$ do
    Apply first-improvement $k = 2$ actions until no further improvement
    if no improvement found then
        Randomly select $a_1$, determine $b_1$, and initialize $\boldsymbol{a}$
        $i \leftarrow 1$
        while $a_{i+1} \neq a_1$ do
            Calculate $\Omega_{b_i}$ as average of $W_{b_{ij}}$ across all edges
            Compute $Z_{b_{ij}}$ for all candidate nodes
            Identify set $X$ where $W_{b_{ij}} \geq 1$
            Select $a_{i+1}$ with probability $P_j$
            Determine $b_{i+1}$ and append $(a_{i+1}, b_{i+1})$ to $\boldsymbol{a}$
            $i \leftarrow i + 1$
            if $a_{i+1} = a_1$ then
                Break
            end if
        end while
        $\boldsymbol{\pi}^{new} \leftarrow$ ApplyAction$(\boldsymbol{\pi}, \boldsymbol{a})$
        if $L(\boldsymbol{\pi}^{new}) < L(\boldsymbol{\pi})$ then
            $\boldsymbol{\pi} \leftarrow \boldsymbol{\pi}^{new}$
            for each edge $(b_i, a_{i+1})$ in $\boldsymbol{a}$ do
                $W_{b_i a_{i+1}} \leftarrow W_{b_i a_{i+1}} + \beta[\exp(\frac{L(\boldsymbol{\pi}) - L(\boldsymbol{\pi}^{new})}{L(\boldsymbol{\pi})}) - 1]$
            end for
            if $L(\boldsymbol{\pi}) < L(\boldsymbol{\pi}^*)$ then
                $\boldsymbol{\pi}^* \leftarrow \boldsymbol{\pi}$
            end if
        else
            Randomly reset $\boldsymbol{\pi}$ to a new starting state
        end if
    end if
    Update $M$ and $Q_{b_i a_{i+1}}$ for all edges in $\boldsymbol{a}$
end while
return $\boldsymbol{\pi}^*$
```

The UTICA algorithm operates in two phases with distinct computational requirements. For heatmap generation, the Graph Convolutional Network processes each subgraph with time complexity $O(m^2)$ where m is the fixed-size problem scale, resulting in overall complexity $O(|L_s| \cdot m^2)$ for $|L_s|$ sampled subgraphs. The reinforcement learning phase operates within the time limit T, with each iteration requiring $O(K)$ operations for K waypoints. The space complexity is dominated by the weight matrix W and visit matrix Q, requiring $O(K^2)$ storage. This computational efficiency makes UTICA practical for UAV trajectory planning applications.

5 EXPERIMENTS

5.1 Experimental Setup

Our simulation employs a 1 km × 1 km ground network with 100-500 randomly distributed IoT devices. The communication range between devices is set to $r_{com} = 100$ meters. The spatial data similarity values $c(i, j)$ between neighboring devices are based on real sensor data collected from 54 sensors over 38 days at the Intel Berkeley Research Lab [16]. For any adjacent devices v_i and v_j, we assign a random similarity value $c(i, j)$ from the interval $[0, 1]$, where values approaching 1 indicate high data correlation and values near 0 represent low correlation. This approach allows us to realistically model the spatial relationships between IoT devices in our experimental scenarios.

We implemented our model using PyTorch 1.0.1.post2 and Python 3.7 on a WSL subsystem with an NVIDIA GeForce RTX 3090 GPU. Training utilized the Adam optimizer with an initial learning rate of 0.001, which was decayed by a factor of 1.01 whenever the validation loss failed to decrease by at least 1% after 5 epochs. The model was evaluated every 5 epochs on a validation set of 10,000 instances. For training, we generated 990,000 instances containing 20 nodes each, with optimal paths determined by the Concorde exact solver as benchmarks.

5.2 Baselines and Metrics

Under identical simulation conditions, we benchmark the proposed algorithm against four baseline methods.

- **Genetic Algorithm** [17]: A metaheuristic that mimics natural selection by generating multiple feasible solutions and evolving them through crossover, mutation, and selection operations to progressively approach the optimal solution.
- **Self-Organizing Map** [18]: An unsupervised learning algorithm that maps the path optimization problem onto a low-dimensional grid structure, using competitive learning mechanisms to identify optimal trajectories.
- **Ant Colony Optimization** [19]: A nature-inspired method based on ant foraging behavior, where multiple agents construct solutions while depositing and following pheromone trails to collectively identify efficient paths.
- **Learning+Greedy** [20]: A hybrid approach that leverages a graph neural network to generate a probability graph, then employs greedy search to sequentially select highest-probability edges when constructing paths.

5.3 Experimental Results

To evaluate the scalability of different approaches, we consider wireless sensor network sizes ranging from 100 to 500 nodes.

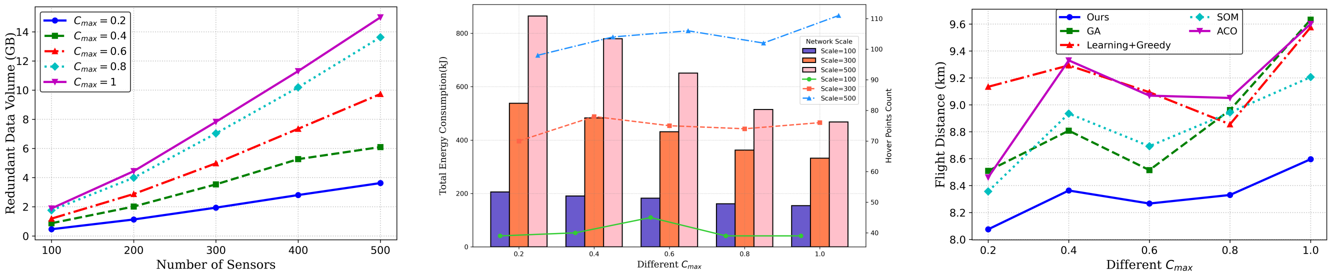
5.3.1 Comparative Analysis for C_{max}. This section evaluates the impact of parameter C_{max} on system performance. As shown in Fig. 3a, the proposed method's effectiveness in suppressing redundant data varies with C_{max}. In small networks with dispersed IoT devices, C_{max} has limited impact on redundant data reduction. However, as network size increases and device density grows, larger C_{max} values significantly reduce redundant data collection during UAV operations. Fig. 3b demonstrates how increasing C_{max} improves the network's redundant data suppression capability, consequently reducing total energy consumption across networks of 100, 300, and 500 nodes. The number of UAV waypoints increases proportionally with network size. Our comparative analysis in Fig. 3c confirms the algorithm's effectiveness in trajectory optimization. With 500 nodes and $C_{max} = 1$, our method achieves the shortest flight distance, outperforming GA by 12.05%, Learning+Greedy by 11.41%, SOM by 7.09%, and ACO by 11.69%.

5.3.2 Comparative Analysis for Energy Efficiency. We evaluate trajectory optimization algorithms and search strategies for UAV energy consumption in networks of 100-500 nodes. As Fig. 4a shows, energy consumption increases with network size for all methods. Our proposed algorithm consistently outperforms GA, Learning+Greedy, SOM, and ACO across all network scales, reducing energy consumption by 7.15%, 7.92%, 4.21%, and 4.66% respectively. These results confirm our method's effectiveness, particularly in large-scale networks. We also compared search strategies including Greedy Search, Beam Search, and our proposed MCTS method. Fig. 4b demonstrates that while all strategies require more energy as network size increases, MCTS consistently achieves the lowest energy consumption, reducing it by 5.68% compared to Greedy Search and 1.28% compared to Beam Search.

5.3.3 Comparative Analysis for Communication Range. We examined how IoT device communication range (50-250 meters) affects system performance. Fig. 5a-c show that increasing communication range significantly enhances redundant data suppression, as wider coverage reduces transmission redundancy. Consequently, larger communication ranges decrease the required UAV waypoints, reducing flight time (Fig. 5b) and optimizing collection efficiency. This improvement extends to total energy consumption (Fig. 5c), where expanded communication capabilities simultaneously minimize redundant data transmission and optimize flight paths, substantially lowering energy requirements.

6 Related Work

Research on data collection in UAV-assisted IoT networks has seen significant growth, with emphasis on simultaneously optimizing collection efficiency and energy consumption. Researchers in [21] proposed a multi-strategy multi-objective ant colony optimization algorithm that jointly designs aerial vehicle trajectories, hover positions, and speeds to minimize task completion time and energy while ensuring safety. Work in [22] introduced a geometric partitioning-based dynamic programming algorithm, optimizing

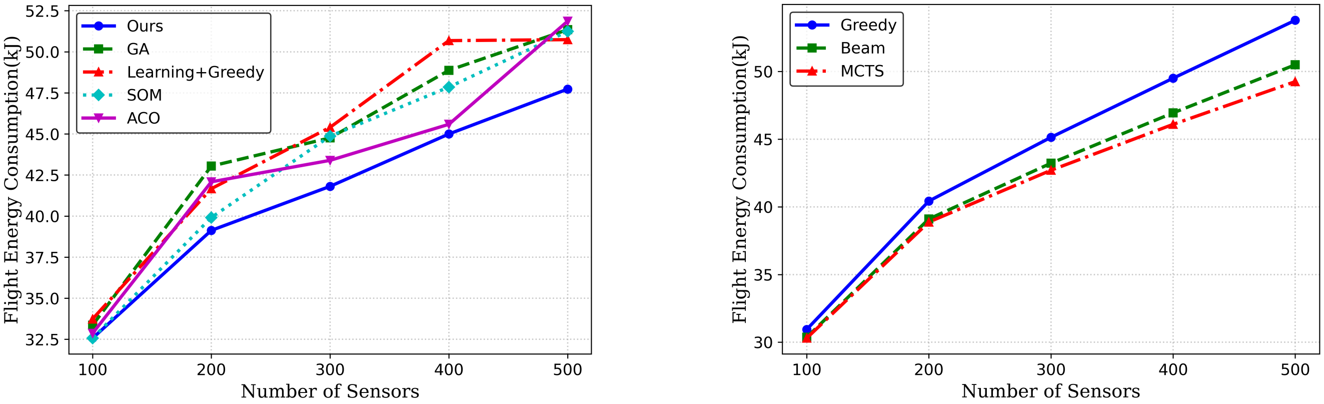


(a) Impact on suppressing redundant data volume.

(b) Impact on Energy Consumption and waypoints.

(c) Impact of different C_{max} on flight distance.

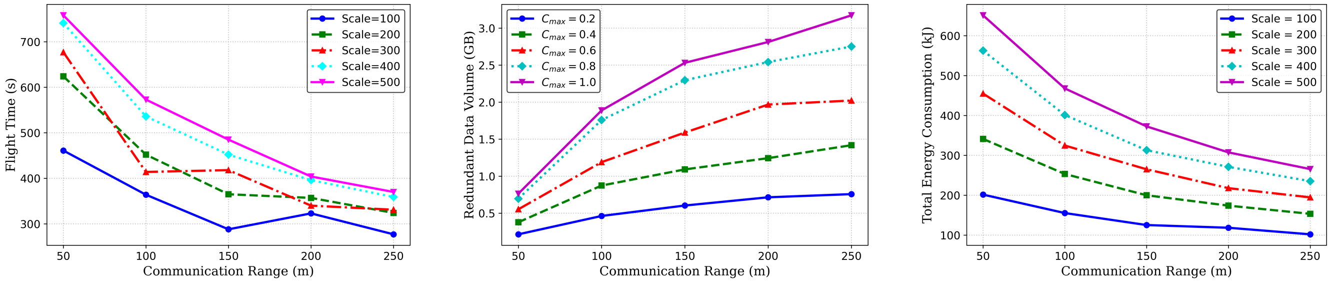
Figure 3: Performance impact of different C_{max} values with $r_{com} = 100$ and a network scale of 500.



(a) Impact of different trajectory optimization algorithms.

(b) Impact of different search strategies on the heatmap.

Figure 4: Performance impact of different optimization algorithms on flight energy consumption with $C_{max} = 1$ and $r_{com} = 100$.



(a) Impact of different communication ranges on flight time.

(b) Impact on the suppression of redundant data volume.

(c) Impact on total energy consumption.

Figure 5: Performance impact of different sensor communication ranges with a network scale of 100.

aerial vehicle paths by partitioning the IoT network and minimizing vehicle numbers under latency constraints. Authors in [23] developed a region of interest partitioning and tracking strategy using mean-shift clustering and Voronoi diagrams to adapt to dynamic user distributions, enhancing network capacity and reducing energy consumption. However, these methods rely primarily on heuristics or mathematical programming, often experiencing exponential growth in computation time when applied to large-scale problems, limiting their practicality in real-world scenarios.

With advancements in machine learning, neural networks have been widely adopted to address these challenges. Research in [24] developed an aerial vehicle data collection framework for multi-cell networks, minimizing operational time and age of information using search-based and graph-based algorithms alongside deep reinforcement learning techniques. Authors in [25] designed an aerial vehicle-assisted wireless charging system for Internet of Things devices, employing constrained Markov decision processes and multi-agent constrained deep reinforcement learning to optimize trajectories while ensuring safety and service quality. Work in [26] proposed deep deterministic policy gradient and hybrid heuristic learning approaches to minimize mobile device energy consumption. Researchers in [27] combined reinforcement learning with imitation learning, training deep neural networks for collaborative aerial vehicle path planning to reduce task completion time and energy consumption. While these machine learning approaches demonstrate strong capabilities in handling complex wireless environments and learning efficient trajectory strategies, they still face limitations in generalization across problems of varying scales.

7 CONCLUSION

This paper presents an energy optimization framework for UAV-assisted IoT data collection that uniquely integrates spatial data correlation analysis with trajectory optimization. By formulating redundancy suppression as a submodular maximization problem and employing Graph Convolutional Networks combined with Monte Carlo Tree Search for path optimization, our approach significantly reduces energy consumption while maintaining high solution quality across various problem scales. Extensive simulations using real-world sensor data demonstrate that our framework achieves up to 35% energy savings compared to conventional approaches. Future work should address dynamic environments with mobile sensors, edge computing integration for real-time decision making, and coordination strategies for multi-UAV deployments to further enhance scalability and resilience in large-scale IoT networks.

References

[1] Maggi Bansal, Inderveer Chana, and Siobhán Clarke. UrbanEnQoSPlace: A deep reinforcement learning model for service placement of real-time smart city IoT applications. *IEEE Transactions on Services Computing*, 16(4):3043–3060, 2022.

[2] Yuanyuan Zeng, Shujie Zhou, and Kai Xiang. Online-offline interactive urban crowd flow prediction toward IoT-based smart city. *IEEE Transactions on Services Computing*, 15(6):3417–3428, 2021.

[3] Dene A Hedges, Justin P Coon, and Gaojie Chen. A continuum model for route optimization in large-scale inhomogeneous multi-hop wireless networks. *IEEE Transactions on Communications*, 68(2):1058–1070, 2019.

[4] Nan Qi, Zanqi Huang, Wen Sun, Shi Jin, and Xiang Su. Coalitional formation-based group-buying for UAV-enabled data collection: An auction game approach. *IEEE Transactions on Mobile Computing*, 22(12):7420–7437, 2022.

[5] Luwei Fu, Zhiwei Zhao, Geyong Min, Wang Miao, Liang Zhao, and Wenjie Huang. Energy-efficient 3-D data collection for multi-UAV assisted mobile crowdsensing. *IEEE Transactions on Computers*, 72(7):2025–2038, 2022.

[6] Xueyuan Wang and M Cenk Gursoy. Resilient path planning for UAVs in data collection under adversarial attacks. *IEEE Transactions on Information Forensics and Security*, 18:2766–2779, 2023.

[7] Avishkar Seth, Alice James, Endrowednes Kuantama, Richard Han, and Subhas Mukhopadhyay. AeroBridge: Autonomous Drone Handoff System for Emergency Battery Service. In *Proceedings of the 30th Annual International Conference on Mobile Computing and Networking*, MobiCom '24, pages 573–587, New York, NY, USA, May 2024. ACM.

[8] Zhidan Liu, Wei Xing, Bo Zeng, Yongchao Wang, and Dongming Lu. Distributed spatial correlation-based clustering for approximate data collection in WSNs. In *Proceedings of the 27th IEEE International Conference on Advanced Information Networking and Applications*, AINA '13, pages 56–63. IEEE, 2013.

[9] Suryansh Sharma, Ashutosh Simha, Venkatesha Prasad, Shubham Deshmukh, Kavin Balaji Saravanan, Ravi Ramesh, and Luca Mottola. BEAVIS: Balloon Enabled Aerial Vehicle for IoT and Sensing. In *Proceedings of the 29th Annual International Conference on Mobile Computing and Networking*, MobiCom '23, pages 1–15, New York, NY, USA, October 2023. ACM.

[10] Zhenjie Guo, Jian Peng, Wenzheng Xu, Weifa Liang, Weigang Wu, Zichuan Xu, Bing Guo, and Yue Ivan Wu. Minimizing redundant sensing data transmissions in energy-harvesting sensor networks via exploring spatial data correlations. *IEEE Internet of Things Journal*, 8(1):512–527, 2020.

[11] Xiuwen Fu, Chang Deng, and Antonio Guerrieri. Low-AoI data collection in integrated UAV-UGV-assisted IoT systems based on deep reinforcement learning. *Computer Networks*, 259:111044, March 2025.

[12] Xueqiang Li, Ming Tao, Shuling Yang, Mian Ahmad Jan, Jun Du, Lei Liu, and Celimuge Wu. AI-empowered intelligent search for path planning in UAV-assisted data collection networks. *IEEE Internet of Things Journal*, 11(21):34492–34503, 2024.

[13] Tong Ding, Ning Liu, Zhong-Min Yan, Lei Liu, and Li-Zhen Cui. An efficient reinforcement learning game framework for UAV-enabled wireless sensor network data collection. *Journal of Computer Science and Technology*, 37(6):1356–1368, 2022.

[14] Yuchen Li, Weifa Liang, Wenzheng Xu, Zichuan Xu, Xiaohua Jia, Yinlong Xu, and Haibin Kan. Data collection maximization in IoT-sensor networks via an energy-constrained UAV. *IEEE Transactions on Mobile Computing*, 22(1):159–174, 2021.

[15] Yong Zeng, Jie Xu, and Rui Zhang. Energy minimization for wireless communication with rotary-wing UAV. *IEEE Transactions on Wireless Communications*, 18(4):2329–2345, 2019.

[16] Intel Berkeley research lab data. Available at http://db.csail.mit.edu/labdata/labdata.html, 2013. Accessed: 2024-07-01.

[17] Sheema Madhusudhanan, Arun Cyril Jose, Jayakrushna Sahoo, and Reza Malekian. PRIMϵ: Novel privacy-preservation model with pattern mining and genetic algorithm. *IEEE Transactions on Information Forensics and Security*, 19:571–585, 2023.

[18] Laura Maria Palomino Mariño and Francisco de Assis Tenorio de Carvalho. Self-organizing maps with adaptive distances for multiple dissimilarity matrices. *Machine Learning*, 113(10):7783–7806, 2024.

[19] Yiqian Wang, Jie Zhu, Haiping Huang, and Fu Xiao. Bi-objective ant colony optimization for trajectory planning and task offloading in UAV-enabled MEC systems. *IEEE Transactions on Mobile Computing*, 23(12):12360–12377, December 2024.

[20] Chaitanya K Joshi, Quentin Cappart, Louis-Martin Rousseau, and Thomas Laurent. Learning the travelling salesperson problem requires rethinking generalization. *Constraints*, 27(1):70–98, 2022.

[21] Riheng Jia, Qiyong Fu, Zhonglong Zheng, Guanglin Zhang, and Minglu Li. Energy and time trade-off optimization for multi-UAV enabled data collection of IoT devices. *IEEE/ACM Transactions on Networking*, 32(6):5172–5187, December 2024.

[22] Shu-Wei Chang, Jian-Jhih Kuo, Mong-Jen Kao, Bo-Zhong Chen, and Qian-Jing Wang. Near-optimal UAV deployment for delay-bounded data collection in IoT networks. In *Proceedings of the IEEE Conference on Computer Communications*, INFOCOM '24, pages 111–120. IEEE, 2024.

[23] Ruide Cao, Jiao Ye, Qian You, Jianghan Xu, Yi Wang, Shiyu Jiang, and Yaomin Li. A novel region-of-interest based UAV planning strategy for mitigating urban peak demand. In *Proceedings of the 24th International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing*, MobiHoc '23, pages 300–301, New York, NY, USA, 2023. ACM.

[24] Cheng Zhan, Han Hu, Jing Wang, Zhi Liu, and Shiwen Mao. Tradeoff between age of information and operation time for UAV sensing over multi-cell cellular networks. *IEEE Transactions on Mobile Computing*, 23(4):2976–2991, 2023.

[25] Zhaolong Ning, Hongjing Ji, Xiaojie Wang, Edith C. H. Ngai, Lei Guo, and Jiangchuan Liu. Joint optimization of data acquisition and trajectory planning for UAV-assisted wireless powered Internet of Things. *IEEE Transactions on Mobile Computing*, 24(2):1016–1030, February 2025.

[26] Bin Dai, Jianwei Niu, Tao Ren, Zheyuan Hu, and Mohammed Atiquzzaman. Towards energy-efficient scheduling of UAV and base station hybrid enabled mobile edge computing. *IEEE Transactions on Vehicular Technology*, 71(1):915–930, 2021.

[27] Jack Farley, Amirahmad Chapnevis, and Eyuphan Bulut. Generalized path planning for collaborative UAVs using reinforcement and imitation learning. In *Proceedings of the 24th International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing*, MobiHoc '23, pages 457–462, New York, NY, USA, 2023. ACM.