Multi-UAV Trajectory Planning for Data Collection Based on Decision Transformer

Ke Zhao[1], Kaixin Li[1], and Steve Jacob Thomas[1]

[1] Kyungpook National University, Daegu, South Korea.
{kezhao, kaixinli}@knu.ac.kr, stevejacobt@gmail.com

Abstract. This paper investigates a three-tier Space-Air-Ground (SAG) uplink communication system comprising High Altitude Unmanned Aerial Vehicles (HAUs), Low Altitude Unmanned Aerial Vehicles (LAUs), and Internet of Things (IoT) nodes. In our system,the LAUs act as relays for data transmission from IoT nodes to the HAUs, and aims to minimize the total mission completion time by optimizing the trajectories of the UAVs while ensuring each IoT node connects to only one UAV per time slot, the total time does not exceed a maximum limit, and all nodes are successfully served. To address this, we propose to apply a decision transformer(DT) algorithm. Through simulations,the DT model accurately predicts stop positions but shows slight deficiencies in predicting movement actions.

Keywords: Trajectory Planning · Data Collection · Decision Transformer

1 Introduction and Background

Enhanced Machine-Type Communications (eMTC) is a critical component of 5G telecommunication systems, and it is expected to continue evolving in beyond 5G (B5G) networks. Its primary goal is to support a wide array of emerging mobile edge computing (MEC) applications across various domains, including transportation, smart cities, smart oceans, forest monitoring, and industrial manufacturing. Task offloading in massive eMTC systems is a fundamental function in MEC. The core objective of task offloading is to allocate user requests for storage and computation to appropriate network entities, which involves transporting data to designated network entities via communication paths provided by the eMTC system.

UAVs, by virtue of their ability to fly close to ground devices and establish low-altitude air-to-ground communication links, can be effectively deployed to hover over areas of interest and collect data from ground-based Internet-of-Things (IoT) networks. This UAV-aided data collection method offers significant energy savings for devices in traditional IoT networks, thereby extending their operational lifetime . However, ensuring the freshness of collected information is crucial, particularly in time-sensitive IoT applications such as environmental

monitoring and safety protection. In these scenarios, data must be transmitted to its destination promptly, as outdated information can result in incorrect actions and potentially cause major disasters .

UAV networks are emerging as a crucial and promising solution for achieving seamless connectivity from ground to space, vital for upcoming 6G communications. SAGINs offer flexibility, scalability, and real-time communication and processing capabilities. Despite their potential, coordinating network resources in mobile, hierarchical, and heterogeneous SAGINs remains a complex challenge. In this context, resource allocation in aerial computing within SAGIN architecture has been explored using both conventional methods and reinforcement learning (RL) approaches.

Traditional RL is grounded in a mature theoretical framework, notably Markov decision processes (MDPs), and is widely applied in gaming, robotics, autonomous systems, and wireless communications. Its classical algorithms, such as Q-learning and Actor-Critic methods, are adaptable to various problems.Offline reinforcement learning (Offline RL) is a method that trains models without real-time interaction with the environment, relying instead on pre-collected, extensive offline data for policy optimization. This approach offers several significant advantages. Firstly, Offline RL can make full use of existing data without the need for costly and time-consuming online sampling, making it especially useful in scenarios where data collection is expensive or real-time interaction is impractical. Secondly, in fields such as healthcare, autonomous driving, and industrial control, real-time experimentation can pose safety risks. Offline RL mitigates these potential dangers by avoiding the need to explore unknown strategies in actual environments, thereby ensuring system safety. The offline RL-Decision Transformer (DT) incorporates self-attention mechanisms to manage unstructured data types like text and complex wireless communication signals. This hybrid approach is well-suited for complex decision-making tasks, particularly those requiring robust handling of large-scale and diverse data in wireless networks. DT's flexibility and scalability across diverse applications and environments make it a promising tool for enhancing communication systems' efficiency and adaptability, addressing challenges like spectrum management, resource allocation, and dynamic network optimization.

2 Related work

In [1], an inter-server computation offloading scheme is proposed to reduce the computational burden of ground Internet of Things (IoT) devices. The scheme adopts an iterative optimization algorithm that combines a heuristic greedy method and a continuous convex approximation technique to improve computational efficiency. [2] introduces a multi-agent proximal policy optimization algorithm combined with a convex optimization-based resource allocation method, aiming to maximize the total rate of AIoE users and meet the needs of the rapidly developing digital environment. [3] studies the joint optimization of UAV three-dimensional trajectory and resource allocation, solving non-convex optimization

problems through an effective iterative algorithm to meet user needs and maximize energy efficiency. Meanwhile, [4] proposes an iterative algorithm combining Lagrangian dual decomposition with a continuous convex approximation method to maximize system energy efficiency by jointly optimizing subchannel selection, uplink transmission power control, and UAV deployment.

Machine learning (ML) methods have been widely used in task offloading and resource allocation, especially in the case of large action spaces or incomplete information. [5] proposed a computation offloading method based on deep reinforcement learning (DRL), which dynamically learns the optimal strategy by leveraging the policy gradient method to handle large action spaces and combining the actor-critic method to accelerate learning. Similarly, [6] proposed a learning-based, queue-aware task offloading and resource allocation algorithm, which addressed challenges such as incomplete information and the curse of dimensionality through decomposition and actor-critic-based task offloading techniques. [7] explored a task offloading algorithm based on deep actor-critic to optimize the decision-making process under incomplete information conditions, further demonstrating the versatility of ML technology in complex network environments.These advances highlight the potential of combining traditional optimization techniques with modern ML methods and advanced UAV network to address the challenges posed by dynamic and complex network environments.

3 System Model and Problem Formulation

3.1 System Model

As shown in Fig. 2, we consider a three-tier SAG uplink communication system consisting of H HAUs, L LAUs, and G IoT nodes. LAUs serve as relays for data transmission from the IoT nodes to the HAU. The flight altitudes of LAUs and HAUs are denoted by h_l and h_h, respectively. We assume thr HAU as an center controller to guide the trajectory of all the LAUs to our fleet of UAVs completes services for all users in the shortest possible time. The total mission completion time $T_{total} = \sum_{n=1} t_n$.

We consider a 3D coordinate system and denote the positions of IoT node i, LAU l and HAU h in different timeslot as $G_i(t_n) = (x_i(t_n), y_i(t_n), 0)$, $L_l(t_n) = (x_l(t_n), y_l(t_n), h_l(t_n))$, $H_h(t_n) = (x_h(t_n), y_h(t_n), h_h(t_n))$, respectively. we consider the speed of each LAU is same and the location of HAU is fixed. The distances of the three uplink channels in a cascading channel, are denoted as $d_{i,l}(t_n)$ and $d_{l,h}(t_n)$, respectively, and calculated by:

$$d_{i,l}(t_n) = \sqrt{(x_l(t_n) - x_i(t_n))^2 + (y_l(t_n) - y_i(t_n))^2 + h_l^2} \tag{1}$$

$$d_{l,h}(t_n) = \sqrt{(x_h(t_n) - x_l(t_n))^2 + (y_h(t_n) - y_l(t_n))^2 + (h_h(t_n) - h_l(t_n))^2} \tag{2}$$

We assume that IoT node i uploads its sensing data using a predefined transmit power p_i, $i \in N$ and we assume the coverage of each UAV is same. The

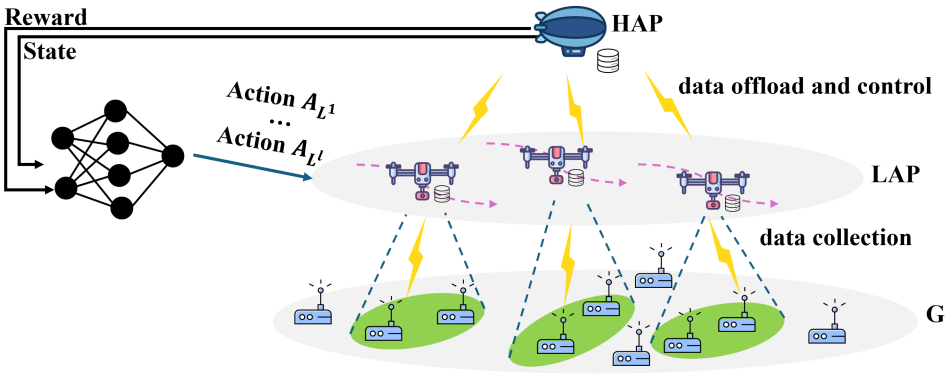


Fig. 1: System Model

instantaneous achievable rate $R_{i,l}((t_n))$ for IoT node i, when transmitting data to the LAU l, is expressed in bits per second (bits/s) as follows:

$$R_{i,l}((t_n)) = a_{i,l}(t_n)B_i \log_2(1 + \frac{p_i \eta}{n_0^2 d_{i,l}(t_n)^2}), \tag{3}$$

where η represents the power gain per unit distance at the reference distance of one meter, the binary variable $a_{i,l}$ is one when LAU l is selected by IoT node i and zero otherwise. p_i denotes the transmit power of IoT node i; B_i represents the bandwidth of IoT node i to LAU l and n_0 signifies the power spectral density of additive white Gaussian noise (AWGN). The transmission delay T_i for IoT nodes to LAUs is expressed as follows:

$$T_i = \frac{D_i}{R_{i,l}(t_n)}, \quad \forall i \in G. \tag{4}$$

where D_i is the data size of IoT node i need to offload. The data rate of L-H link from LAU l to HAU h, denoted by $R_{l,h}$, is defined as follows:

$$R_{l,h}(t_n) = B_{lh} \log_2(1 + \frac{p_l^r G_{l,h} L_s}{k_B T_s B_{lh}}), \tag{5}$$

whereB_{lh} represents the bandwidth of L-H channel; $G_{l,h}$ denotes the antenna power gain; $L_s = (\frac{c}{4\pi d_{l,h}(t_n) f})^2$ is the free-space path loss, where c represents the light speed and f represents the center frequency. Additionally, k_B stands for Boltzmann's constant and T_s represents the system noise temperature. And the channel from LAU to HAU is much stronger, so we ignore the delay of control signals from HAU to LAU.

3.2 Problem Formulation

let $\mathbb{L} = \{L_l(t_n) \mid l \in L, t \in T\}$,Our objective is to ensure that our fleet of UAVs completes services for all users in the shortest possible time. The objective is

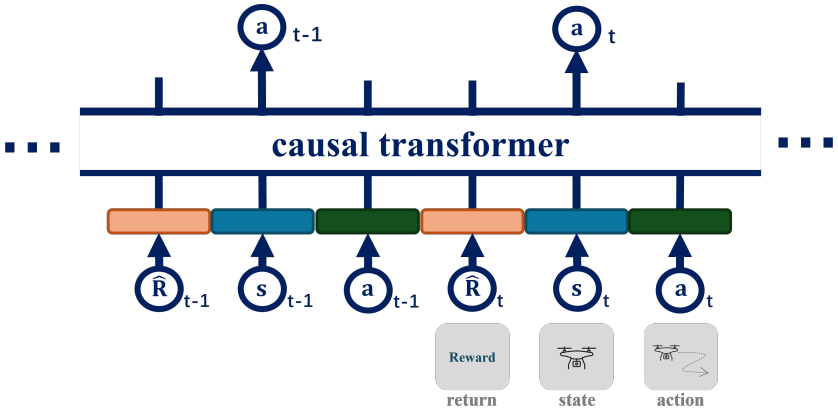


Fig. 2: The architecture of decision transformer

formulated by:

$$\min_{\mathbb{L}} T_{total} \tag{6}$$

$$\textbf{s.t.} \sum_{l=1} a_{i,l}(t_n) \leq 1, \quad \forall i \in I. \tag{7}$$

$$a_{i,l}(t_n) \geq 0, \quad \forall i \in I, l \in L. \tag{8}$$

$$T_{total} \leq T_{max}. \tag{9}$$

$$\sum_{t=1} G_s(t_n) \geq G. \tag{10}$$

$$Eq.(1) - Eq.(5)$$

Eq.(7)-Eq.(8) means that IoT node is connected to only one UAV in a time slot. Eq.(9) means that total time should less than the given maximum time.Eq.(10) means that UAV swarms need to successfully serve all nodes.

4 Decision Transformer for Multi-UAV Trajectory

Algorithm 1 Decision Transformer Pseudocode

```
# R, s, a, t: returns-to-go, states, actions, or timesteps
# transformer: transformer with causal masking (GPT)
# embed_s, embed_a, embed_R: linear embedding layers
# embed_t: learned episode positional embedding
# pred_a: linear action prediction layer

# Main Model
def DecisionTransformer(R, s, a, t):
    # compute embeddings for tokens
    pos_embedding = embed_t(t) # per-timestep (note: not per-token)
    s_embedding = embed_s(s) + pos_embedding
    a_embedding = embed_a(a) + pos_embedding
    R_embedding = embed_R(R) + pos_embedding

    # interleave tokens as (R_1, s_1, a_1, ..., R_K, s_K)
    input_embeds = stack(R_embedding, s_embedding, a_embedding)

    # use transformer to get hidden states
    hidden_states = transformer(input_embeds=input_embeds)

    # select hidden states for action prediction tokens
    a_hidden = unstack(hidden_states).actions

    # predict action
    return pred_a(a_hidden)

# Training Loop
for (R, s, a, t) in dataloader: # dims: (batch_size, K, dim) do
    a_preds = DecisionTransformer(R, s, a, t)
    loss = mean((a_preds - a)^2) # L2 loss
    optimizer.zero_grad(); loss.backward(); optimizer.step()
end

# Evaluation Loop
target_return = 1 # for instance, expert-level return
R, s, a, t, done = [target_return], [env.reset()], [], [1], False
while not done
do
    # sample next action
    action = DecisionTransformer(R, s, a, t)[-1] # for cts actions
    new_s, r, done, _ = env.step(action)
    R = R + [R[-1] - r] # decrement returns-to-go with reward
    s, a, t = s + [new_s], a + [action], t + [len(R)]
    R, s, a, t = R[-K:], ... # only keep context length of K
end
```

Multi-UAV trajectory planning based on Decision Transformer is an innovative approach aimed at generating collision-free and efficient paths for each UAV while considering various constraints and objectives. We formulate the multi-UAV trajectory planning problem as a sequential decision-making problem, where each UAV is represented by its state, action, and reward.

- $S(t_n)$: Set of all LAU locations in t_n,$\{L_1(t_n), ..., L_l(t_n)\}$.
- $A(t_n)$: LAU can choose from five different actions: $A_l = (+x, +y, x, y, 0)$, where $+x$, $+y$, x, or y signifies LAU l shifting right, upward, left, or downward, respectively. The action "0" indicates that LAU l is hovering.
- $R(t_n)$: Our objective is to ensure that our fleet of UAVs completes services for all users in the shortest possible time. So we define the instant rewards are as follows:

$$R(t_n) = G_s(t_n) * w_2 - p_1(t_n) * w_1 \tag{11}$$

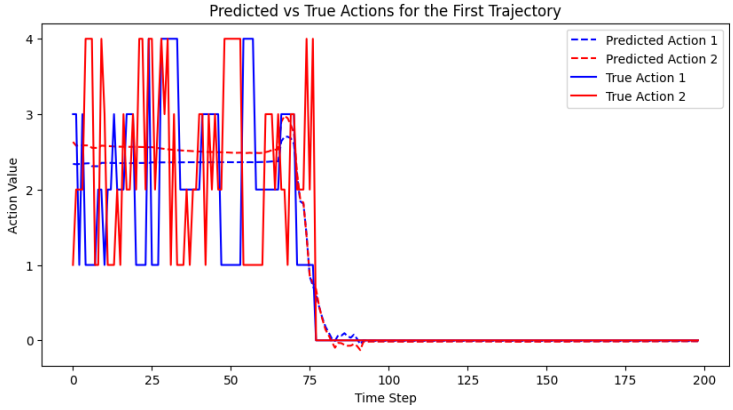
Where w_1 and w_2 are weights, and $p_1(t_n)$ is the penalty for UAV collision or exceeding the boundary.The goal is to find a policy that maximizes cumulative rewards while ensuring safety and efficiency. The Decision Transformer model predicts actions by conditioning on past states, actions, and rewards, treating trajectory optimization as a sequence modeling task. Key components include state embeddings, action embeddings, reward embeddings, and the causal Transformer.

Table 1: Simulation parameters

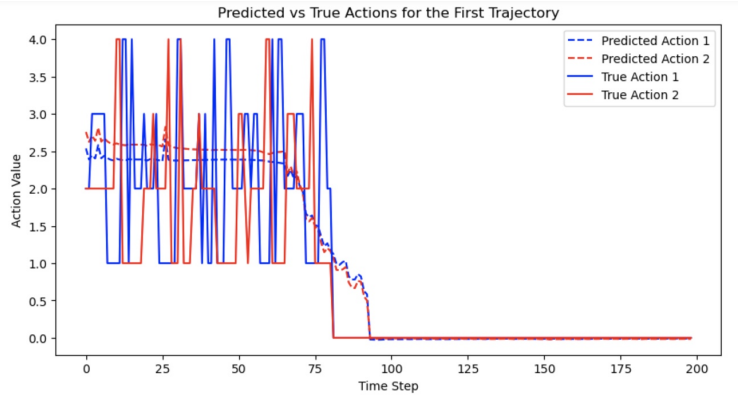
Parameters	**Assumption**
Number of LAUs, L	2
Number of IoT nodes, G	40
Area of the simulation	1000*1000m
LAUs step	100m

In terms of model architecture, the Decision Transformer consists of input embeddings, a Transformer encoder, and an output layer. The input embedding part encodes the state, action, and reward of each UAV, forming an input sequence. The Transformer encoder captures the temporal dependencies between states, actions, and rewards, while the output layer uses a linear layer to predict the next action. The model is trained using supervised learning on historical UAV trajectory datasets, with training steps including data collection, trajectory segmentation, defining the loss function, and optimization.

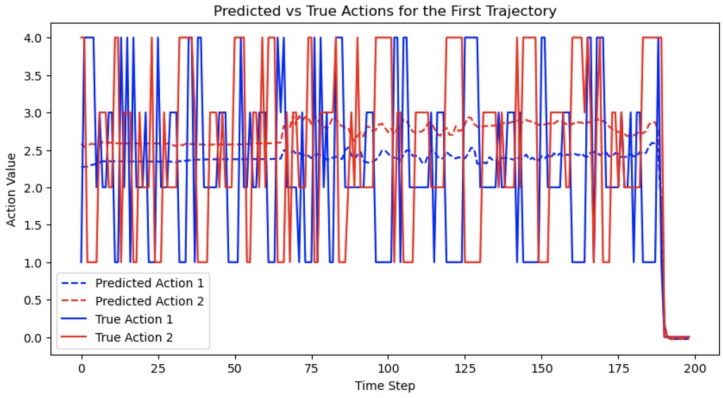
To ensure effective coordination among multiple UAVs, we employ collision avoidance, communication mechanisms, and decentralized control strategies in our approach. Collision avoidance introduces constraints in the reward function to penalize collisions and encourage safe distances between UAVs. The communication mechanism allows UAVs to share their intended trajectories, enabling dynamic adjustments to avoid conflicts. Decentralized control allows each UAV to



(a) scenario 1



(b) scenario 2



(c) scenario 3

Fig. 3: Predicted Action generated by DT

Table 2: The Author Contributions

Author	Project proposal	Paper review	Model conception	Paper writing	Total
Ke Zhao	40%	40%	30%	30%	35%
Kaixin Li	20%	40%	20%	50%	32.5%
Steve Jacob Thomas	40%	20%	50%	20%	32.5%

use a local instance of the Decision Transformer, achieving distributed decision-making while maintaining overall coordination through shared information.

5 Experimental Results

This section presents a numerical analysis of the proposed algorithm. The analysis was performed by running the algorithm on Google CoLab. We consider a three-tier SAG uplink communication system consisting of H HAUs, L LAUs, and G IoT nodes. The simulation parameters used for the analysis are shown in Table 1.

We conducted training on Decision Transformer (DT) over 200 episodes and compared their performance with a data based on pre-trained A2C model The DT model accurately predicts stop positions but shows slight deficiencies in predicting movement actions. This discrepancy may stem from a limited training dataset or the insufficient number of training episodes. Moving forward, we plan to modify the DT actions to discrete actions and increase the training frequency to achieve improved predictive accuracy.

6 Conclusion

This study explores an uplink communication system on the Three-Layer Sky (SAG) consisting of High Altitude Unmanned Aerial Vehicles (HAU), Low Altitude Unmanned Aerial Vehicles (LAU), and Internet of Things (IoT) nodes. In our system, LAUs act as relays transferring data from IoT nodes to HAUs, aiming to minimize total task completion time by optimizing UAV trajectories. To address this challenge, we propose a Decision Transformer (DT) algorithm. Through simulations, the DT model accurately predicts stop positions but shows slight deficiencies in predicting motion actions. Looking ahead, we plan to modify DT actions to discrete actions and increase training frequency to enhance prediction accuracy.

7 Author Contributions

The Author Contributions are shown in Table 2.

References

1. Y. Shi, J. Zhang, Y. Gao, and Y. Xia, "Inter-server computation offloading and resource allocation in multi-drone aided space-air-ground integrated iot networks," *Journal of Communications and Networks*, vol. 24, pp. 324–335, June 2022. 2
2. Y. Gong, H. Yao, D. Wu, W. Yuan, T. Dong, and F. R. Yu, "Computation offloading for rechargeable users in space-air-ground networks," *IEEE Transactions on Vehicular Technology*, vol. 72, pp. 3805–3818, Mar. 2023. 2
3. Z. e. a. Hu, "Joint resources allocation and 3d trajectory optimization for uav-enabled space-air-ground integrated networks," *IEEE Transactions on Vehicular Technology*, vol. 72, pp. 14214–14229, Nov. 2023. 2
4. Z. e. a. Li, "Energy efficient resource allocation for uav-assisted space-air-ground internet of remote things networks," *IEEE Access*, vol. 7, pp. 145348–145362, 2019. 3
5. N. e. a. Cheng, "Space/aerial-assisted computing offloading for iot applications: A learning-based approach," *IEEE Journal on Selected Areas in Communications*, vol. 37, pp. 1117–1129, May 2019. 3
6. H. Liao, Z. Zhou, X. Zhao, and Y. Wang, "Learning-based queue-aware task offloading and resource allocation for space–air–ground-integrated power iot," *IEEE Internet of Things Journal*, vol. 8, pp. 5250–5263, Apr. 2021. 3
7. Z. Wang, Z. Zhou, H. Zhang, G. Zhang, H. Ding, and A. Farouk, "Ai-based cloud-edge-device collaboration in 6g space-air-ground integrated power iot," *IEEE Wireless Communications*, vol. 29, no. 1, pp. 16–23, 2022. 3