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 aim 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 algorithm. Through simulations, we demonstrate the effectiveness of our proposed method in enhancing the performance of the three-tier SAG uplink communication system.

Keywords: Trajectory Planning · Data Collection · Decision Transformer

1 Introduction and Background

Enhanced machine-type communications (eMTC) has been defined under 5G telecommunication systems and is expected to stay evolving in the future beyond (B5G) networks, aiming to support numerous emerging and novel mobile edge computing (MEC) applications in various areas such as transportation, smart city, smart ocean, forest monitoring, and industry manufacture [1]. Task offloading in massive eMTC systems is one of the most fundamental MEC functions. By serving as the core of many relevant applications, the general target of task offloading is to allocate each user request for storage and computation to proper network entities, for which the data shall be transported to the designated network entities via a set of communication paths/channels provided by the eMTC system.

An eMTC system could be deployed in any hostile surroundings where the eMTC nodes may not connect to any terrestrial base station (BS) or fixed access points (AP), but resort to multiple ground vehicles/robots and a set of terrestrial-support-free (TSF) edge terminals that cruise in the air such as unmanned aerial vehicles (UAVs). The ground vehicles/robots could be automated guided vehicles (AGVs), unmanned surface vessels (USVs), or autonomous underwater vehicles (AUVs), depending on various restrictions imposed by the

application environments [2]. The TSF edge terminals, on the other hand, are equipped with certain communication, computation, and storage capacity while cruising on their pre-scheduled trajectories, forming a TSF multi-layer multi-access edge computing (TMMEC) system that can be used to support various data collection-related applications and task offloading.

Research efforts on task offloading under conventional MEC architecture have been extensively reported in the past, mostly focused on solving the issues of clustering the sensor nodes for energy efficiency, trajectory planning of robot(s)/UAVs for cost down, datapath and edge server selection for data transportation, and placement/allocation/dimensioning of system capacity for overall system throughput, while few of them have studied the scenario of TMMEC where the edge terminals are distributed in hierarchy and with high mobility. Particularly, it is critical to reliably transport the information of interest out of the sensing data to the terminals for offloading, mostly located at some high-altitude UAVs (HAUs) and satellites that are equipped with abundant resources and renewable power. However, there have been numerous restrictions and special design requirements for achieving effective networking of the UAVs under TMMEC for the desired reliable data transfer, mainly due to the long-distance fragile wireless links in the presence of high mobility of the UAVs, which concerns the data route formation/selection among the interconnected UAVs. Besides, the energy consumption of the UAVs strongly depends on their trajectories, while optimal/near-optimal planning of the UAV trajectories may incur high computation complexity and should be subject to careful design.

The space-air-ground integrated networks (SAGINs) are recognized as a crucial and promising solution for providing seamless connections, aiming to achieve comprehensive connectivity from ground to space for the upcoming 6G communications. The exploration of SAGINs has attracted significant attention due to their flexibility, scalability in deployment, and capabilities for real-time communication and processing [3] [4] [5] [6]. However, the challenges associated with network coordination and resource management, stemming from the mobile, hierarchical, and heterogeneous nature of SAGINs, remain complex and daunting. In this regard, the issue of resource allocation in aerial computing, leveraging SAGIN architecture, has been extensively investigated using both conventional and reinforcement learning(RL)-based approaches.

Traditional reinforcement learning (RL) benefits from a mature theoretical foundation with well-established frameworks like Markov decision processes (MDPs), enabling its widespread application across domains such as gaming, robotics, autonomous systems, and wireless communications. Its diverse range of classical algorithms, including Q-learning and Actor-Critic methods, ensures adaptability to various problem contexts in these fields. In contrast, a Decision Transformer (DT) integrates self-attention mechanisms to effectively manage unstructured data types such as text, and intricate wireless communication signals [7]. This hybrid approach enhances its suitability for complex decision-making tasks, particularly in scenarios requiring robust handling of large-scale and diverse data in wireless networks. DT's flexibility and scalability across

diverse applications and environments make it promising for advancing the efficiency and adaptability of communication systems, addressing challenges such as spectrum management, resource allocation, and dynamic network optimization.

2 Related work

In [8], an inter-server computation offloading scheme is proposed to reduce computation overhead for ground Internet of Things (IoT) devices. This scheme employs an iterative optimization algorithm that integrates heuristic greedy methods with successive convex approximation techniques. [9] introduces the use of a multi-agent proximal policy optimization algorithm alongside convex optimization-based resource allocation, aiming to maximize the total rate for artificial intelligence of everything (AIoE) users. [10] explores the joint optimization of UAV 3D trajectory and resource allocation, with the goal of meeting user requirements while maximizing energy efficiency, through an effective iterative algorithm for solving the non-convex optimization problem. [11] proposes an iterative algorithm that combines Lagrangian dual decomposition with successive convex approximation methods, aimed at maximizing system energy efficiency via the joint optimization of sub-channel selection, uplink transmission power control, and UAV deployment.

ML-based methods have been extensively applied to task offloading and resource allocation, especially in scenarios characterized by large action spaces or incomplete information. [12] introduced a DRL-based computing offloading approach that dynamically learns the optimal policy by leveraging the policy gradient method for large action spaces and the actor-critic method to accelerate learning. [13] proposed a learning-based, queue-aware task offloading and resource allocation algorithm that addresses challenges such as incomplete information and the curse of dimensionality, utilizing decomposition and actor-critic-based task offloading techniques. [14] explored a queue-aware deep actor-critic (Q-DAC)-based task offloading algorithm to optimize decision-making under conditions of incomplete information.

Nonetheless, most of the research works mentioned previously assume a fixed SAGIN topology, despite dynamic task requests, without considering the changes in link connectivity that result in temporal network variability. In response, recent research efforts have addressed the challenges posed by high mobility by modeling the time-varying network topology using time-expanded graphs (TEG) [15] [16] [17] [18]. [15] presented a deterministic satellite network transmission approach utilizing TEG, which integrates deterministic spatiotemporal routing, redundant coding, and multi-path scheduling. It aims to enhance communication performance by predicting communication opportunities and optimizing routing paths. [16] introduced a time-graph method for energy-limited flow routing in satellite networks, aiming to maximize data flow while maintaining energy efficiency. [17] proposed a transceiver resource allocation scheme for satellite networks based on the TEG, facilitating efficient allocation by representing resources as virtual nodes and edges. [18] introduced the multi-functional

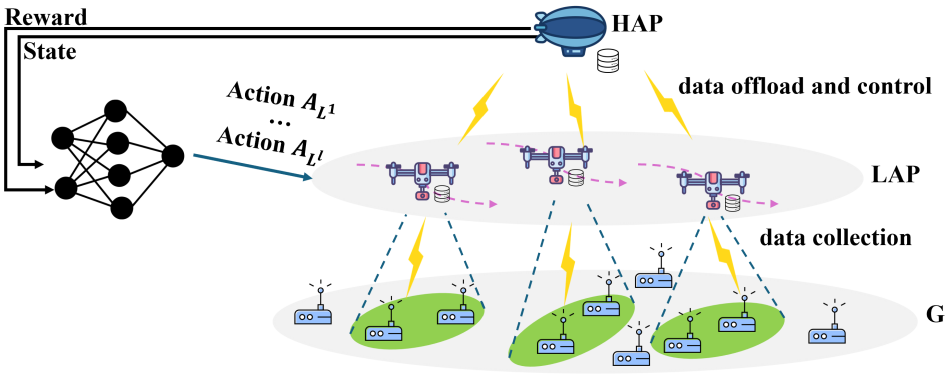


Fig. 1: System Model

TEG (MF-TEG) to address the complexities of satellite communication networks in a single-source to single-destination scenario, by jointly modeling communication, storage, and computation capabilities through the decomposition of nodes into virtual components.

3 System Model and Problem Formulation

3.1 System Model

As shown in Fig. 1, 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 the HAU as a center controller to guide the trajectory of all the LAUs to our fleet of UAVs to complete 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 to be the same and the location of the 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 the same. The 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. 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 formulated by:

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

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

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

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

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

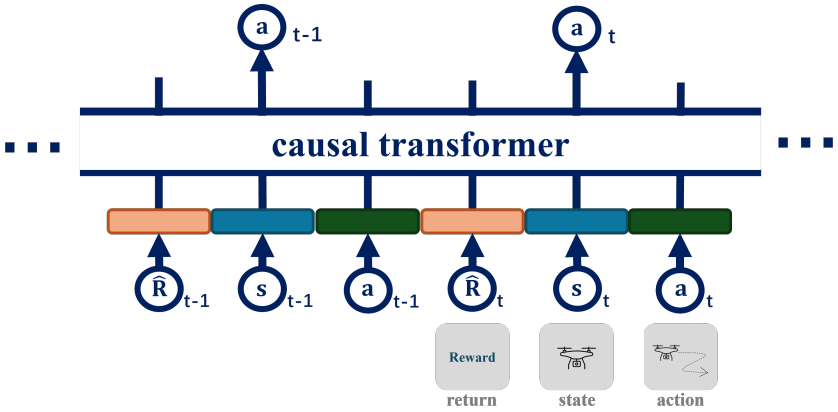


Fig. 2: The architecture of decision transformer

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

Eq.(7)-Eq. (8) means that the IoT node is connected to only one UAV in a time slot. Eq.(9) means that the total time should be 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

Multi-UAV trajectory planning based on a 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 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.

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
```

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 use a local instance of the Decision Transformer, achieving distributed decision-making while maintaining overall coordination through shared information. The architecture of the decision transformer is shown in Fig. 2 and the pseudocode of the DT algorithm is shown in **Algorithm 1**.

5 Experimental Results

5.1 Simulation setting

This section presents a numerical analysis of the proposed algorithm. The analysis was performed by running the algorithm on Google CoLab. The simulation parameters used for the analysis are shown in Table 1.

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

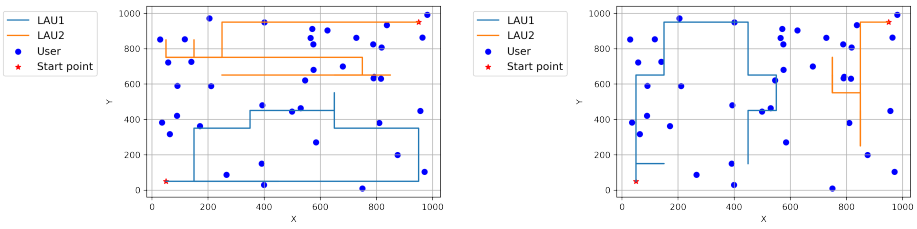
Our dataset is derived from a pre-trained Advantage Actor-Critic (A2C) model [19]. It comprises 7,000 distinct entries, each detailing the real-time state of a drone during a sequence of 200 consecutive actions, the subsequent action taken, and the corresponding reward received.

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 hl and hh, respectively. We assume the HAU as a center controller to guide the trajectory of all the LAUs to our fleet of UAVs to complete services for all users in the shortest possible time. The total mission completion time.

We consider performance comparisons using two benchmarks. First, in Benchmark 1, we guide the trajectory using the A2C algorithm. Second, in Benchmark 2, we use the Proximal Policy Optimization (PPO) method [20].

5.2 Numerical Results

In this part, DT, A2C and PPO algorithms are used to optimize the trajectories of multiple UAVs respectively. First, we used the DT algorithm to optimize the trajectories of multiple UAVs, and we assumed two scenarios. In the first case, the coverage range of the UAV is within 120 meters of the radius of the UAV, as shown in Fig. 3a. In the second case, the coverage range of the UAV is within 200 meters of the radius of the UAV, as shown in Fig. 3b. The results show that each IoT node can successfully connect to a drone per time slot, and all nodes can be successfully served without exceeding the maximum time limit.



(a) **(b)**

Fig. 3: DT for different coverage: 120m(a),200m(b)

Fig. 4 shows the result of using the A2C algorithm to optimize the UAV's trajectory. It can be seen from the results that the A2C algorithm can also complete the service for all IoT nodes, and in the same two coverage areas as the DT algorithm, the A2C algorithm uses fewer steps than the DT algorithm. However DT algorithm leverages pre-collected expert trajectories for learning, which may make it more data-efficient than the A2C algorithm, which typically requires extensive interaction with the environment to learn a policy, DT algorithm does not rely on real-time interaction with the environment, and its training process may be more stable and less susceptible to environmental noise. The A2C algorithm, on the other hand, may encounter issues with balancing exploration and exploitation during training, leading to instability.

Fig. 5 shows the results of optimizing UAV trajectory using the PPO algorithm. It can be seen from the results that A2C algorithm y can also complete the service for all IoT nodes, but in two different coverage areas, the A2C algorithm uses more steps than the DT algorithm and A2C algorithm.

In this part, we use DT, A2C, and PPO algorithms respectively to optimize the trajectories of multiple UAVs. First of all, the proposed DT algorithm is close to the A2C algorithm in performance, but the DT algorithm has better

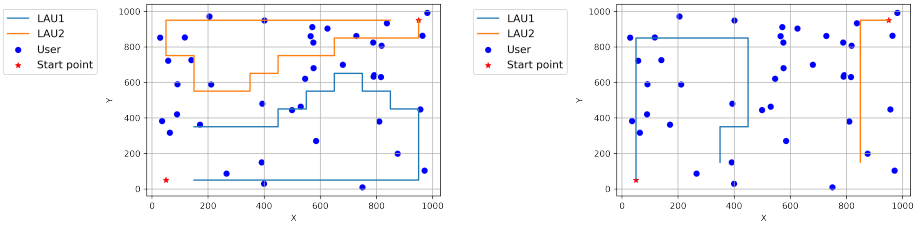


Fig. 4: A2C for different coverage: 120m(a),200m(b)

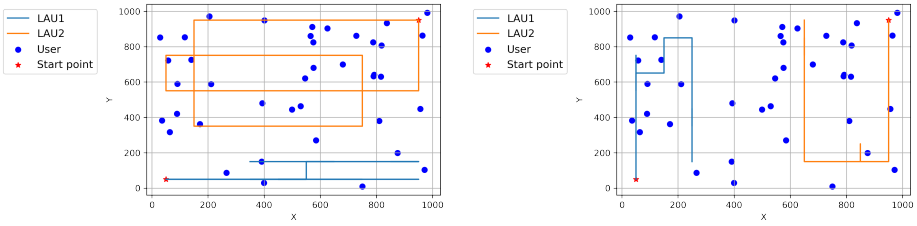


Fig. 5: PPO for different coverage: 120m(a),200m(b)

data efficiency and stability. The performance of our proposed DT algorithm is also better than that of the PPO algorithm.

6 Conclusion

In this paper, multi-UAV trajectory planning for data collection using the decision transformer was investigated. The study focused on UAV Trajectory Planning. Specifically, our problem seeks to minimize total time 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. The results demonstrated that using the Decision Transformer method achieves excellent performance in UAV trajectory planning, and its performance is close to that of the dominant Actor Critic (A2C) method and better than that of the near end Strategy optimization (PPO) method.

7 Author Contributions

The Author Contributions are shown in Table 2.

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%

References

1. S. Mao, S. He, and J. Wu, "Joint uav position optimization and resource scheduling in space-air-ground integrated networks with mixed cloud-edge computing," *IEEE Systems Journal*, vol. 15, no. 3, pp. 3992–4002, 2021. 1
2. L. Zhang, W. Abderrahim, and B. Shihada, "Heterogeneous traffic offloading in space-air-ground integrated networks," *IEEE Access*, vol. 9, pp. 165462–165475, 2021. 2
3. S. A. Huda and S. Moh, "Survey on computation offloading in uav-enabled mobile edge computing," *Journal of Network and Computer Applications*, vol. 201, p. 103341, 2022. 2
4. S. Rahim and L. Peng, "Intelligent space-air-ground collaborative computing networks," *IEEE Internet of Things Magazine*, vol. 6, no. 2, pp. 76–80, 2023. 2
5. X.-T. Li, S. Xu, Z.-P. Zhao, Z.-Y. Li, D.-A. Li, and J.-M. Zhao, "A survey on computing offloading in satellite-terrestrial integrated edge computing networks," in *2023 15th International Conference on Communication Software and Networks (ICCSN)*, pp. 172–182, 2023. 2
6. J. Mo, K. Zhao, and L. Peng, "Joint deployment of lau and hau for hierarchical space-air-ground communications," in *2023 International Conference on Networking and Network Applications (NaNA)*, pp. 133–137, 2023. 2
7. L. Chen, K. Lu, A. Rajeswaran, K. Lee, A. Grover, M. Laskin, P. Abbeel, A. Srinivas, and I. Mordatch, "Decision transformer: Reinforcement learning via sequence modeling," in *Advances in Neural Information Processing Systems* (M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, eds.), vol. 34, pp. 15084–15097, Curran Associates, Inc., 2021. 2
8. 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. 3
9. 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. 3
10. 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. 3
11. 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
12. 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

13. 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
14. 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
15. X. e. a. Jiang, "Spatio-temporal routing, redundant coding and multipath scheduling for deterministic satellite network transmission," *IEEE Transactions on Communications*, vol. 71, pp. 2860–2875, May 2023. 3
16. K. Shi, H. Li, and L. Suo, "Temporal graph based energy-limited max-flow routing over satellite networks," in *2021 IFIP Networking Conference (IFIP Networking)*, (Espoo and Helsinki, Finland), pp. 1–3, 2021. 3
17. P. Wang and X. e. a. Zhang, "Time-expanded graph-based resource allocation over the satellite networks," *IEEE Wireless Communications Letters*, vol. 8, pp. 360–363, Apr. 2019. 3
18. W. Liu, H. Yang, and J. Li, "Multi-functional time expanded graph: A unified graph model for communication, storage, and computation for dynamic networks over time," *IEEE Journal on Selected Areas in Communications*, vol. 41, pp. 418–431, Feb. 2023. 3
19. M. Babaeizadeh, I. Frosio, S. Tyree, J. Clemons, and J. Kautz, "Reinforcement learning through asynchronous advantage actor-critic on a gpu," *arXiv preprint arXiv:1611.06256*, 2016. 8
20. J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017. 9