drones

MDPI

Article

Joint Trajectory Design and Resource Optimization in UAV-Assisted Caching-Enabled Networks with Finite Blocklength Transmissions

Yang Yang * **and Mustafa Cenk Gursoy**

Department of Electrical Engineering and Computer Science, Syracuse University, Syracuse, NY 13244, USA; mcgursoy@syr.edu
* Correspondence: yyang82@syr.edu

Abstract: In this study, we design and analyze a reliability-oriented downlink wireless network assisted by unmanned aerial vehicles (UAVs). This network employs non-orthogonal multiple access (NOMA) transmission and finite blocklength (FBL) codes . In the network, ground user equipments (GUEs) request content from a remote base station (BS), and there are no direct connections between the BS and the GUEs. To address this, we employ a UAV with a limited caching capacity to assist the BS in completing the communication. The UAV can either request uncached content from the BS and then serve the GUEs or directly transmit cached content to the GUEs. In this paper, we first introduce the decoding error rate within the FBL regime and explore caching policies for the UAV. Subsequently, we formulate an optimization problem aimed at minimizing the average maximum end-to-end decoding error rate across all GUEs while considering the coding length and maximum UAV transmission power constraints. We propose a two-step alternating optimization scheme embedded within a deep deterministic policy gradient (DDPG) algorithm to jointly determine the UAV trajectory and transmission power allocations, as well as blocklength of downloading phase, and our numerical results show that the combined learning-optimization algorithm efficiently addresses the considered problem. In particular, it is shown that a well-designed UAV trajectory, relaxing the FBL constraint, increasing the cache size, and providing a higher UAV transmission power budget all lead to improved performance.

Keywords: unmanned aerial vehicle (UAV); non-orthogonal multiple access (NOMA); finite blocklength (FBL) codes; content caching

check for updates

Citation: Yang, Y.; Gursoy, M.C. Joint Trajectory Design and Resource Optimization in UAV-Assisted Caching-Enabled Networks with Finite Blocklength Transmissions. *Drones* **2024**, *8*, 12. https://doi.org/10.3390/drones8010012

Academic Editor: Emmanouel T. Michailidis

Received: 15 October 2023
Revised: 15 December 2023
Accepted: 19 December 2023
Published: 4 January 2024

1. Introduction

Recently, unmanned aerial vehicles (UAVs) have been extensively utilized across various domains, such as enhancing wireless coverage and contributing to the development of smart cities, as noted in previous studies [1,2]. The utilization of UAVs is recognized as a promising technique in numerous 5G applications, owing to their inherent characteristics, which include rapid mobility, cost-effectiveness, and extended airtime, as highlighted in the literature [3]. To be more precise, low-altitude UAVs can be exploited by wireless communication networks for swift deployment and enhanced mobility flexibility, as outlined in [4]. These advantages imply the growing importance of UAV-enabled communication systems in upcoming wireless networks.

However, the rapid evolution of 5G networks has led to a significant surge in wireless communication demands. Data traffic congestion is mostly attributed to the repeated downloads of a few popular contents. To mitigate this bottleneck, edge caching technology has emerged as a promising solution, enabling edge servers to cache frequently accessed contents. In certain scenarios, UAVs can act as edge servers to serve ground user equipment (GUEs) and cache popular contents. In [5], the authors have explored the joint optimization of UAV deployment, caching placement, and user association in UAV-assisted cellular networks, with the goal of maximizing the mean opinion score (MOS) for all users within the cell.

Non-orthogonal multiple access (NOMA) is considered as a promising technology that has been extensively studied in communication systems with relays, demonstrating remarkable effectiveness in enhancing the performance of overloaded networks, as discussed in [6]. Moreover, NOMA has become increasingly popular for its capability to significantly enhance spectral efficiency, making it a potent candidate for enabling low-latency communications by serving multiple users simultaneously. When NOMA transmission is integrated with UAVs, especially when employing successive interference cancellation (SIC) at the receiver, it is anticipated to further enhance the wireless propagation environment. The performance comparisons between NOMA and orthogonal multiple access (OMA) in short-packet communications, under the finite blocklength (FBL) regime, has been explicitly analyzed in [7]. Additionally, the study in [8] has demonstrated a method to maximize the sum rate by optimally determining the UAV's position and power allocations when NOMA transmission is adopted.

Ultra-reliable and low latency communication (URLLC) is a pivotal component of 5G networks and is primarily focused on delivering mission-critical services, as highlighted in [9]. URLLC often involves the use of short packets under the FBL regime, which is of great importance in reducing transmission delays. Consequently, FBL codes necessitate significant modifications in wireless communication system design and performance analysis. In other words, the traditional concept of Shannon's information capacity, applicable under the assumption of infinite blocklength, becomes inapplicable, meaning the decoding error probability under the FBL regime can no longer be neglected. In [10], the authors have presented an analysis of the transmission rate when employing FBL codes in an additive white Gaussian noise (AWGN) channel, explicitly delving into the decoding error probability. Furthermore, in [11], the authors have conducted an analysis of globally optimal resource allocation for URLLC with FBL codes.

1.1. Related Work

Existing research has touched on several aspects. For example, authors in [12] have constructed a UAV-assisted downlink transmission model, considering a two-user NOMA scenario with energy and caching capacity constraints on the UAV. In another work, authors in [13] have investigated UAV deployment and content placement in a cache-enabled multi-UAV network, aiming to minimize the user request delays. Additionally, a comparison of the achievable effective capacity between the two-user NOMA and its OMA counterpart under delay quality-of-service (QoS) constraints within the FBL regime has been explored by the authors in [14]. Moreover, the authors in [15] analyzed the performance of rate-splitting multiple access (RSMA) in a multi-user downlink wireless network where a UAV-assisted BS serves multiple GUEs simultaneously. They also conducted network optimization in the presence of imperfect channel state information (CSI), considering both FBL and infinite blocklength (IBL) regimes. However, there is no UAV trajectory incorporated in this paper, and the content caching introduced in our paper distinguishes our work from [15] significantly.

In [16], the authors have investigated a UAV-enabled secure communication with FBL codes aiming to maximize the average effective secrecy rate (AESR) by jointly designing the UAV's trajectory and transmit power. This paper provides a comprehensive analysis of UAV communications using FBL codes, investigating reliability and latency aspects, but NOMA transmissions and the caching policy are not considered. Another URLLC-enabled UAV relay system, which is similar to our system model, is investigated in [17], where the authors have studied the joint location and blocklength allocation for the UAV relay system with URLLC requirements. However, this paper only considers a 2D scenario with only one robot (GUE), and no caching at the UAV. Last but not least, a more recent work in [18] proposes a novel framework for efficient UAV deployment and resource allocation for Internet-of-Things (IoT) devices in URLLC service scenarios, where multiple UAVs are deployed as aerial BSs to provide URLLC communication for IoT devices. The objective in [18] is to minimize the system's average transmit power by simultaneously optimizing the scheduling and association of IoT devices, power control, bandwidth allocation, and the deployment of UAVs. We notice that the system can be further improved by utilizing NOMA transmissions and the caching policy, which is one of the main contributions in this paper. We have summarized the aforementioned related works in Table 1, below.

Table 1. Summary of related research.

Paper	Considered System	Novelty in Our Paper
[12]	A UAV-assisted downlink transmission network	FBL regime
[13]	A cache-enabled multi-UAV network	FBL regime
[14]	A downlink two-user NOMA network	UAV and caching
[15]	A multi-user downlink wireless network	UAV trajectory designs and caching
[16]	A UAV-enabled secure communication system	NOMA and caching
[17]	A URLLC-enabled UAV relay system	3D scenario, multiple GUEs and caching
[18]	A UAV-assisted IoT network	NOMA and caching

Our proposed framework with a caching-enabled UAV using NOMA transmissions and FBL codes provides several advantages: 1. Reduced dependency: caching popular contents at the UAV allows it to locally serve GUEs without requiring it to connect to the BS, which reduces the dependency on the BS and mitigating potential challenges like network congestion; 2. Enhanced spectral efficiency: NOMA transmissions enable the UAV to serve multiple UEs simultaneously in the same frequency band, resulting in improved spectral efficiency and more effective use of the available spectrum resources; 3. Improved reliability: FBL codes are designed to account for the finite blocklength regime, optimizing the use of coding resources for reliable communication; 4. Optimized resource allocation: the joint optimization of UAV trajectory, power allocation, and content caching allows for efficient resource utilization. In summary, the integration of caching-enabled UAVs with NOMA transmissions and FBL codes contributes to a more efficient, low-latency, and reliable wireless network.

1.2. Motivations and Contributions

We note that numerous studies have been conducted in the field of NOMA transmissions considering the infinite blocklength coding regime. However, in practical scenarios, all wireless transmissions are performed using finite blocklength codes, (and if the finite coding length is sufficiently large, then infinite blocklength assumption can be invoked as a good approximation). In other words, considering the FBL regime in wireless transmissions is practically more relevant and accurate, especially when the code lengths are relatively short due to latency requirements. Recently, NOMA has attracted much interest as a multiple access technique that allows multiple GUEs to share the same time-frequency resources. NOMA transmissions not only improve spectral efficiency, allowing for more efficient use of the available bandwidth, but also support low-latency communication and high throughput by allowing simultaneous transmissions. More importantly, this is advantageous in applications that require real-time communication, and will benefit more from FBL codes. In our considered system model, caching at UAV allows frequently requested contents to be stored locally, reducing the need to retrieve data from a distant data center, which minimizes back-haul traffic and can lead to more efficient use of the network resources. The proposed caching policy in this paper can dynamically adjust the cached contents at the UAV based on GUE preferences or geographical locations, and such a flexibility allows for adaptive and efficient content delivery strategies. Overall, the combination of FBL codes, NOMA transmissions, and caching at the UAV enable low-latency communications while making efficient use of resources.

In this paper, we combine the FBL regime with NOMA and content caching in a UAV-assisted network, with the goal of minimizing the maximum end-to-end decoding error probability when multiple GUEs are involved. Unlike our previous work in [19], where we aim to find the optimal resource allocation at the UAV only for a fixed UAV position, in this paper we comprehensively investigate both the optimal UAV trajectory design and the solutions of the optimal power allocations at the UAV as well as the optimal duration of the downlink (DL) phase. A two-step alternating optimization scheme embedded within a deep deterministic policy gradient (DDPG) algorithm has been constructed to jointly determine the UAV trajectory, transmission power allocations as well as the blocklength of DL phase, to alleviate data traffic burden and enhance the reliability in URLLC. Our main contributions in this paper are summarized as follows:

1. We describe and analyze the UAV-assisted downlink NOMA tranmissions with FBL codes and content caching.
2. We investigate the end-to-end decoding error probability at the GUE and the signal-to-noise ratio (SNR) or signal-to-interference-plus-noise ratio (SINR) in transmissions.
3. We construct a caching policy for the UAV.
4. We develop a two-step alternating optimization scheme-embedded DDPG algorithm to minimize the average maximum end-to-end decoding error rate among all GUEs under both coding length and maximum UAV transmission power constraints.

The remainder of this paper is organized as follows. In Section 2, we start with presenting the system model and conducting an analysis of the FBL regime as well as the SINR when employing NOMA transmissions. We subsequently delve into the determination of end-to-end decoding error probabilities and the caching policy at the UAV. Moving on to Section 3, we formulate an optimization problem with the objective of minimizing the average maximum end-to-end decoding error rate across all GUEs. This optimization problem takes into account both the coding length and the maximum transmission power constraints at the UAV. To address this problem, we construct a two-step alternating optimization scheme embedded DDPG algorithm. In Section 4, we present the results of our simulations, and analyze the performance of our approach. Finally, in Section 5, we summarize the paper and draw conclusions.

2. System Model

In this paper, we study a downlink system model consisting of a base station (BS), a UAV, and a set of N GUEs, represented by $\mathcal{N} = 1, 2, \ldots, N$, as depicted in Figure 1. Each of these communication terminals is equipped with a single antenna. Considering the unpredictable and complex nature of wireless communication environments, such as natural landscapes or densely populated urban areas, we make the assumption that all direct communication links from the BS to the GUEs are unavailable. Consequently, we deploy a UAV with limited cache capacity to serve the GUEs by utilizing NOMA transmissions in the FBL regime. The UAV is capable of moving on a trajectory at a fixed altitude. Throughout this research, we assume that all communication channels remain quasi-static and unchanged within a transmission frame. In other words, the parameters optimized for the current transmission frame, such as transmission power allocations at the UAV, are effective within that frame.

We denote the UAV's cache size as C_{uav}, and we consider a total of C contents that can be requested by the GUEs, with the size of the c-th content designated as I_c bits. If the requested content is available in the UAV's cache, it is transmitted to the GUE without involving the BS. Otherwise, the UAV requests this content from the BS before the transmission from the UAV to the GUE starts.

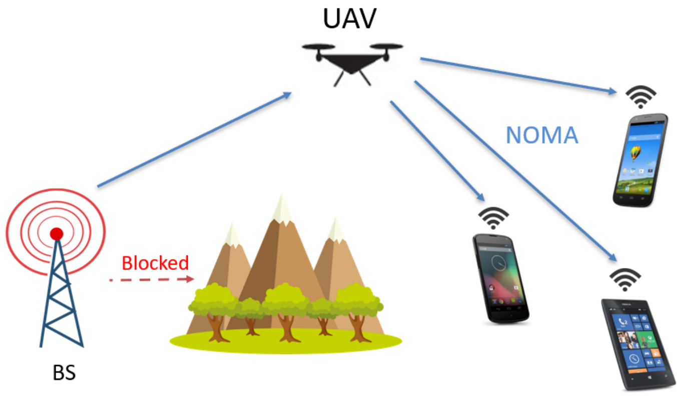


Figure 1. An illustration of the considered network.

The key parameters of the system and their notations are summarized in Table 2, and the abbreviations are summarized in Table 3.

Table 2. Summary of parameters and notations.

N	Number of GUEs
I	Number of considered time frames
T	Completion latency constraint
M	Blocklength constraint of a frame
L	Length of the caching list
C	Number of contents
T_{syb}	Duration of a transmission symbol
m_1	Symbol length of the downlink (DL) phase
m_2	Symbol length of the requesting phase
ε	Decoding error probability
D_n	Size of the content requested by GUE n
O_c	Popularity of the content c
h_n	Fading vector between the UAV and GUE n
h_{uav}	Fading vector between the UAV and the BS
γ	Signal-to-noise ratio (SNR)/signal-to-interference-plus-noise ratio (SINR)
ρ_n	Power allocation factor for the n-th GUE
η	Additive white Gaussian noise (AWGN)
σ^2	Power of the AWGN
$P_{\max}$	Budget for transmission power at the UAV
$X_{c,n}$	Request indicator for GUE n and content c
Y_c	Caching indicator for content c
Z_c	Request indicator for content c

2.1. FBL Transmission with Caching

In this paper, the duration of a transmission symbol is denoted as T_{syb} seconds, and therefore a delay limitation of T seconds corresponds to $M = T/T_{\mathrm{syb}}$ symbols. To be more specific, T seconds, or equivalently M symbol durations, set the maximum frame length for completing the requested content or task. Within a frame, two phases exist: a requesting phase spanning m_2 symbols and a downlink (DL) transmission phase encompassing m_1 symbols, as depicted in Figure 2. In this study, we introduce $X_{c,n,i} \in \{0,1\}$ to indicate the request of the n-th GUE ($X_{c,n,i} = 1$ implies that the n-th GUE is requesting content c

in the i-th frame). The size of the requested content for the n-th GUE in the i-th frame is $D_{n,i} = \sum_{c=1}^{C} X_{c,n,i} I_c$ bits. It is worth noting that within each frame, each GUE is restricted to requesting only one content, e.g., $\sum_{c=1}^{C} X_{c,n,i} = 1, \forall n \in \mathcal{N}$. The UAV first checks its cache: if the requested content is cached, there is no need to consult the BS; otherwise, the content must be downloaded from the BS. After checking its cache for all requested contents in the i-th frame, the UAV proceeds to download all the uncached but requested contents from the BS through a wireless link in the requesting phase, which spans $m_2 T_{\mathrm{syb}}$ seconds. Subsequently, in the DL transmission phase lasting $m_1 T_{\mathrm{syb}}$ seconds, the UAV transmits all the requested contents to the GUEs through NOMA transmissions. It is evident that the total service time for each content request is constrained by $m_1 + m_2 = M$. Following the approach in [10], the coding rate R in the FBL regime is approximated as

$$R \approx \log_2(1+\gamma) - \sqrt{\frac{V}{m}} \frac{Q^{-1}(\varepsilon)}{\ln 2}, \tag{1}$$

where ε represents the probability of decoding error, m is the blocklength, γ stands for the SNR or SINR at the receiver, Q^{-1} is the inverse function of $Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^{\infty} e^{-\frac{t^2}{2}} dt$, and V is the channel dispersion, defined as $V = 1 - (1+\gamma)^{-2}$.

Table 3. Summary of abbreviations.

UAV	Unmanned aerial vehicle
GUE	Ground user equipment
NOMA	Non-orthogonal multiple access
SIC	Successive interference cancellation
FBL	Finite blocklength
URLLC	Ultra-reliable and low latency communication
AWGN	Additive white Gaussian noise
DDPG	Deep deterministic policy gradient
SNR/SINR	Signal-to-noise ratio/signal-to-interference-plus-noise ratio
BS	Base station
DL	Downlink
DRL	Deep reinforcement learning
MDP	Markov decision process
DQN	Deep Q-network
DNN	Deep neural network

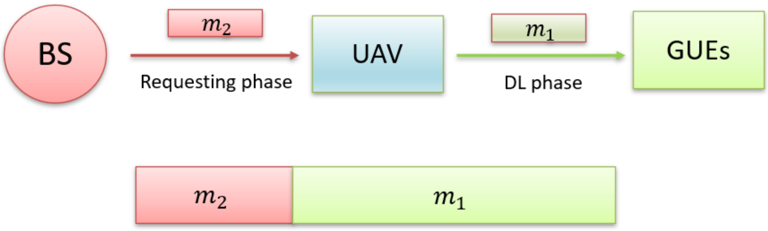


Figure 2. System topology and frame structure.

In this paper, we introduce the notation $Y_{c,i} \in \{0,1\}$ as the caching indicator. Specifically, $Y_{c,i} = 1$ signifies that content c has been cached at the UAV during the i-th frame. Additionally, we define $Z_{c,i}$ as the requesting indicator, as follows:

$$Z_{c,i} = \begin{cases} 1 & \text{when } \sum_{n=1}^{N} X_{c,n,i} \geq 1; \\ 0 & \text{when } \sum_{n=1}^{N} X_{c,n,i} = 0. \end{cases} \tag{2}$$

In particular, $Z_{c,i} = 1$ indicates that content c has been requested in the i-th frame by one or more GUEs. Subsequently, during the i-th frame, the size of all the requested but uncached contents is $D_{\text{uav,i}} = \sum_{c=1}^{C} Z_{c,i}(1 - Y_{c,i})I_c$ bits. Given that the target coding rate in the request phase is $R_{\text{uav,i}} = \frac{D_{\text{uav,i}}}{m_2}$, the decoding error probability of the UAV in the i-th frame during the requesting phase can be expressed as

$$\varepsilon_i^{\text{UAV}} \approx Q\left(\sqrt{\frac{m_2}{V_{\text{uav,i}}}}\left(\log_2(1 + \gamma_{\text{uav,i}}) - \frac{D_{\text{uav,i}}}{m_2}\right)\log_e 2\right). \tag{3}$$

Taking $R_{n,i} = \frac{D_{n,i}}{m_1}$ as the desired achievable coding rate for the n-th GUE in the i-th frame, the decoding error probability during the DL phase can be formulated as follows

$$\varepsilon_{n,i} \approx Q\left(\sqrt{\frac{m_1}{V_{n,i}}}\left(\log_2(1 + \gamma_{n,i}) - \frac{D_{n,i}}{m_1}\right)\log_e 2\right). \tag{4}$$

It is important to notice that, operating within the FBL regime, the blocklength of each frame is constrained by M, and the receiver's decoding error probability is not negligible.

2.2. UAV Trajectory and SINR in Transmissions

In the i-th frame, the position of the UAV at the given altitude z_{uav} is denoted by $(x_i^{\text{uav}}, y_i^{\text{uav}}, \text{z}_{\text{uav}})$, where z_{uav} is assumed to be constant in this paper, and the locations of GUEs are fixed and represented by $(x_1, y_1, 0), (x_2, y_2, 0), \ldots, (x_n, y_n, 0), \ldots, (x_N, y_N, 0)$, respectively. Therefore, the distance between the UAV and the n-th GUE in the i-th frame can be calculated by $d_{n,i} = \sqrt{(x_i^{\text{uav}} - x_n)^2 + (y_i^{\text{uav}} - y_n)^2 + \text{z}_{\text{uav}}^2}$. The positions of UAV over different frames constitute the UAV trajectory in the entire considered period.

Referring to Equations (3) and (4), it is evident that SINR plays a substantial role in influencing the decoding error probability. Therefore, in this section, we specifically investigate the SINR under various transmission scenarios.

During the requesting phase, the UAV is receiving the required data from the BS. Given our assumption of quasi-static channels, we consider the channels to remain unchanged within a frame. As a result, the SNR for the UAV during the requesting phase in the i-th frame is determined by the following expression:

$$\gamma_{\text{uav,i}} = \rho_{\text{uav}}|h_{\text{uav,i}}|^2, \tag{5}$$

where $h_{\text{uav,i}}$ represents the channel coefficient between the UAV and the BS, which varies depending on changing UAV positions. Additionally, ρ_{uav} is the ratio of the transmission power at BS to the noise power, calculated as $\frac{P_{\text{BS}}}{\sigma^2}$, with P_{BS} as the transmission power from the BS to the UAV, and σ^2 denoting the noise power of the AWGN.

During the DL phase, the UAV transmits combined signals to all GUEs based on the NOMA principle. Consequently, the signal received by each GUE in the i-th frame can be described as follows:

$$y_{n,i} = h_{n,i} \sum_{k=1}^{N} \sqrt{P_{\max}\rho_{k,i}} x_{k,i} + \eta, \forall n \in \mathcal{N}, \tag{6}$$

where $x_{k,i}$ and $\rho_{k,i}$ stand for the message and the power allocation factor of the k-th GUE in the i-th frame, respectively. $P_{\max}$ represents the constraint or budget for transmission power at the UAV, and η denotes the AWGN, e.g., $\eta \sim \mathcal{CN}(0, \sigma^2)$. Additionally, $h_{n,i}$ is the channel coefficient between the UAV and the n-th GUE in the i-th frame, and this also varies depending on UAV positions. Note that $\sum_{k=1}^{N} \rho_{k,i} = 1$.

To implement the SIC within the NOMA technique, we initiate a reordering process for all GUEs based on their channel quality at the start of each frame. In the i-th frame, the N GUEs are arranged in ascending order of their corresponding channel quality, specifically, $|h_{1,i}| \leq |h_{2,i}| \leq \ldots, \leq |h_{N,i}|$. The GUE with the weakest channel is designated as the first GUE, while the one with the strongest channel holds the position of the last GUE. Adhering to the SIC principle, for the n-th GUE (where $1 \leq n \leq N$), the signals from all the previous $n-1$ GUEs are decoded first. Subsequently, these decoded signals are subtracted from the superposed received signal. Consequently, the SINR for the n-th GUE to decode its own signal in the i-th frame can be described as follows:

$$\gamma_{n,i} = \frac{|h_{n,i}|^2 P_{\max} \rho_{n,i}}{\sum_{t=n+1}^{N} |h_{n,i}|^2 P_{\max} \rho_{t,i} + \sigma^2}. \tag{7}$$

In the FBL regime, it is essential to note that the SIC errors cannot be ignored. This is because the n-th GUE must first successfully decode the signals from the preceding $n-1$ GUEs before proceeding to decode its own signal. In cases where SIC fails for any GUE, the decoding process for that GUE will also be unsuccessful. Therefore, it is crucial to investigate the error rate associated with the decoding of signals from other GUEs. The SINR for the n-th GUE in decoding the signal of the k-th GUE (where $k \leq n-1 < N$) in the i-th frame can be expressed as follows:

$$\gamma_{n,k,i} = \frac{|h_{n,i}|^2 P_{\max} \rho_{k,i}}{\sum_{t=k+1}^{N} |h_{n,i}|^2 P_{\max} \rho_{t,i} + \sigma^2}. \tag{8}$$

The first GUE can directly decode its own signal by treating the signals from all other GUEs as interference, as there is no SIC being performed at GUE 1. On the other hand, the last GUE conducts a total of $N-1$ SIC processes, and the calculation of its SINR becomes relatively straightforward if all the SIC processes are successful:

$$\gamma_{N,i} = \frac{|h_{N,i}|^2 P_{\max} \rho_{N,i}}{\sigma^2}. \tag{9}$$

2.3. End-to-End Decoding Error

The primary goal of our study is to minimize the average maximum end-to-end decoding error rate for all GUEs while adhering to both coding length and maximum UAV transmission power constraints within the specified time frames. In this section, we delve into the analysis of the end-to-end decoding error probability for GUEs within a particular frame. For the n-th GUE in the i-th frame, we explore two distinct scenarios: whether the content that is being requested has already been cached at the UAV or not.

In the first scenario, where the content requested by the n-th GUE has been cached at the UAV, the end-to-end decoding error probability $\epsilon_{n,i}^{\mathrm{CA}}$ is composed of two main elements: the error probability $\epsilon_{n,k,i}^{\mathrm{SIC}}$ associated with decoding signals from other GUEs when employing SIC, and the error probability $\epsilon_{n,i}$ when decoding its own signal. This is represented as follows:

$$\begin{aligned} \epsilon_{n,i}^{\mathrm{CA}} &= 1 - \prod_{k=1}^{n-1} (1 - \epsilon_{n,k,i}^{\mathrm{SIC}})(1 - \epsilon_{n,i}) \\ &\overset{(a)}{\approx} \sum_{k=1}^{n-1} \epsilon_{n,k,i}^{\mathrm{SIC}} + \epsilon_{n,i}. \end{aligned} \tag{10}$$

Approximation (a) is applicable here because the decoding error probabilities are in the order of 10^{-5} in the considered ultra-reliable communication scenario. Consequently, any terms involving two or more error multiplications can be safely disregarded.

We then turn our attention to the second scenario, where the end-to-end decoding error probability $\epsilon_{n,i}^{\text{UN}}$ for the n-th GUE in the i-th frame consists of three main components: the error probability ϵ_i^{UAV} when decoding the downloaded content from the BS at the UAV, the error probability $\epsilon_{n,k,i}^{\text{SIC}}$ when decoding signals from other GUEs using SIC, and the error probability $\epsilon_{n,i}$ when decoding its own signal. In this scenario, we have

$$\begin{aligned}\epsilon_{n,i}^{\text{UN}} &= 1-(1-\epsilon_i^{\text{UAV}})\prod_{k=1}^{n-1}(1-\epsilon_{n,k,i}^{\text{SIC}})(1-\epsilon_{n,i}) \\ &\overset{(b)}{\approx} \epsilon_i^{\text{UAV}}+\sum_{k=1}^{n-1}\epsilon_{n,k,i}^{\text{SIC}}+\epsilon_{n,i}.\end{aligned} \tag{11}$$

Approximation (b) holds here for the same reason as in approximation (a). By considering both cases, we can provide a more comprehensive description of the end-to-end decoding error rate for the n-th GUE in the i-th frame, denoted as $\epsilon_{n,i}^{\text{tot}}$:

$$\epsilon_{n,i}^{\text{tot}} = \sum_{c=1}^{C} X_{c,n,i}(1-Y_{c,i})\epsilon_i^{\text{UAV}}+\sum_{k=1}^{n-1}\epsilon_{n,k,i}^{\text{SIC}}+\epsilon_{n,i}. \tag{12}$$

In Equation (12), we can calculate ϵ_i^{UAV} by using (3), and $\epsilon_{n,i}$ can be determined from (4). Regarding $\epsilon_{n,k,i}^{\text{SIC}}$, it can be computed as follows:

$$\epsilon_{n,k,i}^{\text{SIC}} \approx Q\left(\sqrt{\frac{m_1}{V_{n,k,i}}}\left(\log_2(1+\gamma_{n,k,i})-\frac{D_{k,i}}{m_1}\right)\log_e 2\right), \tag{13}$$

where $\gamma_{n,k,i}$ can be calculated using the equation in (8), and $V_{n,k,i}$ is defined as $V_{n,k,i} = 1-(1+\gamma_{n,k,i})^{-2}$.

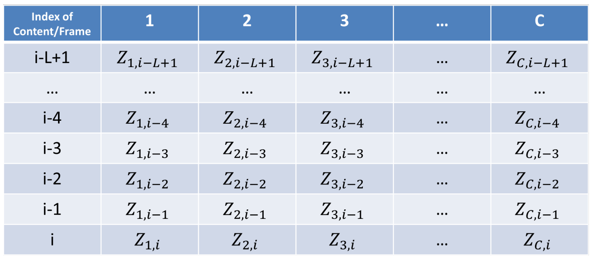
Note that ϵ_i^{UAV}, $\epsilon_{n,i}$ and $\epsilon_{n,k,i}^{\text{SIC}}$ will change if the position of UAV varies, which is due to the fact that all the corresponding SNR/SINRs will be different when the UAV position as well as the channel coefficient changes.

2.4. Caching Policy

In this section, we present our UAV caching policy. Our primary objective with this caching approach is to store the most popular and frequently requested contents. To achieve this, we maintain a caching list that records all the request information from the past L frames on the UAV. Before the start of the $i+1$-th frame, the UAV will remove the request information of the $i-L$-th frame and incorporate the request information of the i-th frame into the caching list, as illustrated in Figure 3. Subsequently, the UAV calculates the popularity of each content, denoted as $O_{c,i}$, which represents the popularity of content c in the i-th frame, and it can be calculated as follows:

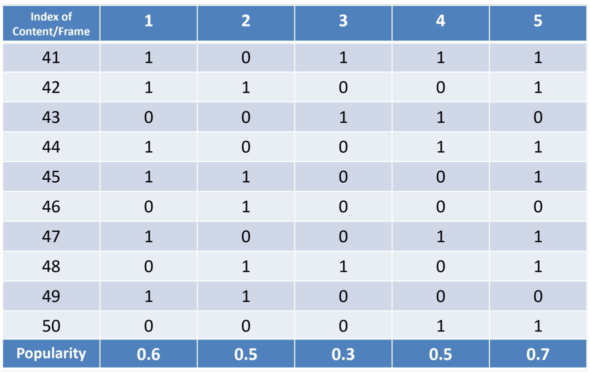
$$O_{c,i} = \frac{\sum\limits_{l=1}^{L} Z_{c,i-l+1}}{L}. \tag{14}$$

Once the popularity of all contents has been determined, the UAV proceeds to cache contents in descending order of popularity until it reaches the cache size limit C_{uav}. Following this, the UAV updates the caching indicator $\{Y\}$ for use in the $i+1$-th frame. As an example, Figure 4 provides an illustration of a caching list for $i=50$, $L=10$, and $C=5$. Assuming that all contents have the same size and the UAV's cache can only accommodate 2 contents, by the end of the 50th frame, the UAV will cache content 1 and content 5.



Index of Content/Frame	1	2	3	…	C
i-L+1	$z_{1,i-L+1}$	$z_{2,i-L+1}$	$z_{3,i-L+1}$	…	$z_{C,i-L+1}$
…	…	…	…	…	…
i-4	$z_{1,i-4}$	$z_{2,i-4}$	$z_{3,i-4}$	…	$z_{C,i-4}$
i-3	$z_{1,i-3}$	$z_{2,i-3}$	$z_{3,i-3}$	…	$z_{C,i-3}$
i-2	$z_{1,i-2}$	$z_{2,i-2}$	$z_{3,i-2}$	…	$z_{C,i-2}$
i-1	$z_{1,i-1}$	$z_{2,i-1}$	$z_{3,i-1}$	…	$z_{C,i-1}$
i	$z_{1,i}$	$z_{2,i}$	$z_{3,i}$	…	$z_{C,i}$

Figure 3. An illustration of the caching list.



Index of Content/Frame	1	2	3	4	5
41	1	0	1	1	1
42	1	1	0	0	1
43	0	0	1	1	0
44	1	0	0	1	1
45	1	1	0	0	1
46	0	1	0	0	0
47	1	0	0	1	1
48	0	1	1	0	1
49	1	1	0	0	0
50	0	0	0	1	1
Popularity	0.6	0.5	0.3	0.5	0.7

Figure 4. An example of the caching list with popularity.

3. Minimization of Maximum Error Probability

In this section, we first formulate and analyze the minimization of the average maximum error rate in the considered network and then propose a two-step alternating optimization scheme embedded within a DDPG algorithm to tackle the proposed problem.

3.1. Problem Formulation

In this paper, our objective is to minimize the average maximum end-to-end decoding error rate among all GUEs within a given period/number of frames, by jointly determining the UAV trajectory $\{(x_i^{\text{uav}}, y_i^{\text{uav}})\}$, UAV transmission power allocation factors $\{\rho_{n,i}\}$, and the length of the DL phase $\{m_{1,i}\}$ subject to the coding length and UAV transmission power constraints. Consequently, the global optimization problem is formulated, as follows:

$$\begin{aligned}
\mathbf{P0:} \quad & \underset{\{(x_i^{\text{uav}}, y_i^{\text{uav}})\}, \{\rho_{n,i}\}, \{m_{1,i}\}}{\textbf{Minimize}} \quad \frac{1}{I} \sum_{i=1}^{I} \max_{n \in \mathcal{N}} \{\epsilon_{n,i}^{\text{tot}}\} \qquad (15) \\
\textbf{s. t.} \quad & \sum_{n=1}^{N} \rho_{n,i} = 1, \quad \forall i \in \mathcal{I} \\
& m_{1,i} + m_{2,i} = M, \quad \forall i \in \mathcal{I} \\
& m_{1,i}, m_{2,i} \in \mathbb{Z}, \quad \forall i \in \mathcal{I}
\end{aligned}$$

where $\mathcal{I} = \{1, 2, \ldots, I\}$ is the set of considered frames.

In **P0**, the first two constraints are the UAV transmission power limitation and the maximum coding length within each frame, respectively. Solving the non-convex problem **P0** directly is quite challenging due to the strongly coupled parameters $\{(x_i^{\text{uav}}, y_i^{\text{uav}})\}, \{\rho_{n,i}\}, \{m_{1,i}\}$ and highly non-linear objective function. In order to address this, we propose a DDPG-based deep reinforcement learning method embedded with a two-step alternating optimization scheme.

3.2. Deep Deterministic Policy Gradient Reinforcement Learning

In this section, we introduce and analyze the main structure of the DDPG reinforcement learning to address the UAV trajectory design in **P0**. DDPG stands out as a prominent deep reinforcement learning (DRL) algorithm that combines aspects of both value-based and policy-based RL techniques. It operates within the actor–critic framework, where the actor network is responsible for selecting actions based on the current environment state, while the critic network evaluates the value of these chosen actions. Both networks are trained simultaneously using the same set of experiences gathered by the agent during its interactions with the environment.

To tackle the challenge of sample correlation in reinforcement learning, DDPG employs a replay buffer to store experiences and randomly samples from this buffer during network updates. Derived from the deterministic policy gradient theorem for Markov decision processes (MDPs) with continuous action spaces, DDPG trains the networks using a stochastic gradient descent with mini-batches and updates the target networks through a soft update mechanism. The target network and replay buffer play significant roles in enhancing stability and sample efficiency.

It is evident that the UAV positions in problem **P0** are continuous, rendering the use of the deep Q-network (DQN) algorithm infeasible, as it is designed for discrete actions. The policy gradient method is sub-optimal in the considered wireless communications, as it suffers from slow convergence. Therefore, we introduce a DDPG-based algorithm to address problem **P0** with respect to the UAV trajectory design. DDPG is an off-policy actor–critic algorithm that operates without a particular system model. It is capable of learning policies in high-dimensional, continuous action spaces [20].

The action space, state space, and reward function of the proposed DDPG reinforcement learning agent are defined as follows:

3.2.1. Action Space

In this paper, we assume that the UAV remains at a fixed altitude, limiting its movement to the horizontal x-y plane. The action space in our proposed DDPG reinforcement learning consists of the UAV movements $\boldsymbol{A} = \{\alpha_v, \alpha_\phi\}$, i.e.,

$$a^i = \boldsymbol{A}^i, \tag{16}$$

where α_v represents the current speed relative to the maximum speed, with values ranging from 0 to 1, and α_ϕ serves as a steering signal that specifies the desired yaw (rotation) angle (normalized by the maximum yaw angle), ranging from -1 to 1. Note that, in this paper, we assume the maximum speed of the UAV is constrained by V_{uav} and the maximum yaw angle is limited by Φ_{uav}.

3.2.2. State Space

The state space in this DRL consists of the horizontal position of the UAV in the previous frame $\boldsymbol{U}^i = \{(x_{i-1}^{\text{uav}}, y_{i-1}^{\text{uav}})\}$, the angle between the previous direction of movement of the UAV and the x-axis $\boldsymbol{\phi}^i = \phi_{i-1}$, the GUEs' current request list $\boldsymbol{X}^i = \{X_{c,n,i}\}$, and the caching list generated from the previous frame $\boldsymbol{Y}^i = \{Y_{c,i}\}$, i.e.,

$$s^i = \{\boldsymbol{U}^i, \boldsymbol{\phi}^i, \boldsymbol{X}^i, \boldsymbol{Y}^i\}. \tag{17}$$

With any given action in the i-th frame, the x-y position of the UAV in the i-th frame can be computed by

$$x_i^{\text{uav}} = x_{i-1}^{\text{uav}} + \alpha_v V_{\text{uav}} M T_{\text{syb}} \cos(\phi_{i-1} + \alpha_\phi \Phi_{\text{uav}}), \tag{18}$$

$$y_i^{\text{uav}} = y_{i-1}^{\text{uav}} + \alpha_v V_{\text{uav}} M T_{\text{syb}} \sin(\phi_{i-1} + \alpha_\phi \Phi_{\text{uav}}). \tag{19}$$

Here, ϕ_i is the angle between the current direction of movement of the UAV and the x-axis in the i-th frame. In this paper, we assume $\phi_0 = 0$, and thereby we have

$$\phi_i = \phi_{i-1} + \alpha_\phi \Phi_{\text{uav}}. \tag{20}$$

3.2.3. Reward Function

In **P0**, the objective is a long time average minimization problem, and it is quite challenging to directly obtain the optimal solutions considering such a long time duration. Therefore, in this paper, we equivalently minimize the maximum end-to-end decoding error probability during each frame. Consequently, we construct the reward function with the objective to minimize the maximum end-to-end decoding error rate among all GUEs in any given frame i, i.e.,

$$r^i = \mathcal{R} - V \log(\max_{\forall n \in \mathcal{N}} \{\epsilon_{n,i}^{\text{tot}}\}), \tag{21}$$

where $\mathcal{R}$ and V are constants to balance the reward. $\{\epsilon_{n,i}^{\text{tot}}\}$ is obtained when the embedded two-step alternating optimization subroutine is completed with the given UAV position $(x_i^{\text{uav}}, y_i^{\text{uav}})$, which is provided by the updated state space.

Based on the above definitions, we propose a DRL-based algorithm according to the DDPG algorithm described in [20]. In this section, we aim to solve **P0** without considering the optimization of UAV transmission power allocations $\{\rho_{n,i}\}$ and the length of the DL phase $\{m_{1,i}\}$, which will be addressed via an embedded two-step optimization subroutine introduced later. Such a proposed DDPG-based algorithm can be deployed at the BS which can collect all the required information about the channel states and apply the policy to all served GUEs and UAV.

The primary distinction between traditional DDPG and our proposed algorithm in this paper lies in the integration of the optimization subroutine aiming to optimize UAV transmission power allocation factors $\{\rho_{n,i}\}$ and the length of the DL phase $\{m_{1,i}\}$.

The DDPG reinforcement learning comprises two essential components within the learning agent: (a) an actor network, responsible for determining the action based on the current state, and (b) a critic network, tasked with assessing the action chosen using the reward feedback from the environment. These networks are represented as $\mu(s|\psi^{\mu})$ and $Q(s, a|\psi^{Q})$, with neural network weights denoted as ψ^{μ} and ψ^{Q}, respectively. The DDPG reinforcement learning algorithm includes three sequential steps.

The initial step involves gathering experience through interactions within the environment. Using the current network state s^i, the actor network produces actions related to the UAV movement. The embedded two-step optimization subroutine determines UAV transmission power allocation factors $\{\rho_{n,i}\}$ and the length of the DL phase $\{m_{1,i}\}$. Subsequently, this joint action is executed at the UAV. The corresponding reward r^i and the subsequent state s^{i+1} are observed from the environment. The transition of the state information, represented as (s^i, a^i, r^i, s^{i+1}), is stored within the experience replay memory to facilitate the training of both the actor and critic networks.

The next step involves the training of the actor and critic networks using the accumulated experience. To prevent potential issues of divergence stemming from deep neural networks (DNNs), a random minibatch of transitions is extracted from the experience replay memory, breaking the correlation between experiences. The training of the critic network focuses on minimizing the loss function

$$\mathcal{L}(\psi^{Q}) = \frac{1}{N_{\text{b}}} \sum_{\iota=1}^{N_{\text{b}}} \Big(y_{\iota} - Q(s_{\iota}, a_{\iota}|\psi^{Q}) \Big)^2, \tag{22}$$

where N_{b} denotes the size of minibatch, and

$$y_{\iota} = r_{\iota} + \zeta Q'(s_{\iota+1}, \mu'(s_{\iota+1}|\psi^{\mu'})|\psi^{Q'}), \tag{23}$$

where ζ is the discount factor, $\mu'(s|\psi^{\mu'})$ denotes the actor target network with weight $\psi^{\mu'}$, and $Q'(s,a|\psi^{Q'})$ is the critic target network with weight $\psi^{Q'}$. Subsequently, the actor network is trained according to the policy gradient

$$\nabla_{\psi^{\mu}} \approx \frac{1}{N_{\mathrm{b}}}\sum_{\iota=1}^{N_{\mathrm{b}}} \nabla_{a} Q(s_{\iota},a|\psi^{Q})|_{s=s_{\iota},a=\mu(s_{\iota})}\nabla_{\theta^{\mu}}\mu(s_{\iota}|\psi^{\mu})|_{s_{\iota}}. \tag{24}$$

The final step is the update of target networks. To maintain the stability of network training, the actor and critic target networks are updated softly:

$$\psi^{\mu'} = \delta\psi^{\mu} + (1-\delta)\psi^{\mu'}, \psi^{Q'} = \delta\psi^{Q} + (1-\delta)\psi^{Q'}, \tag{25}$$

where $\delta \in (0,1]$ represents the update ratio of the target network.

3.3. Two-Step Alternating Optimization

In the previous subsection, we have introduced a DDPG method to tackle the problem of UAV trajectory designs in **P0**. When the UAV trajectory is given, i.e., its location in each frame is known, we can transform **P0** into following problem **P1**:

$$\begin{aligned}
\textbf{P1:}\ & \underset{\{\rho_{n,i}\},m_{1,i}}{\textbf{Minimize}}\ \max_{\forall n\in\mathcal{N}}\{\epsilon_{n,i}^{\mathrm{tot}}\} \\
\textbf{s. t.}\ & \sum_{n=1}^{N}\rho_{n,i} = 1, \\
& m_{1,i} + m_{2,i} = M, \\
& m_{1,i}, m_{2,i} \in \mathbb{Z}.
\end{aligned} \tag{26}$$

Solving **P1** in the i-th frame enables us to jointly optimize the transmission power allocation factors $\{\rho_n\}$ for GUEs and the length of the DL phase m_1 with the objective aiming to minimize the maximum end-to-end decoding error rate among all GUEs while adhering to the coding length and UAV transmission power constraints.

In this subsection, we propose a two-step alternating optimization subroutine to demonstrate how to attain the optimal UAV transmission power allocation factors $\{\rho_{n,i}\}$ and the optimal length of the DL phase $m_{1,i}$ when the UAV position is fixed in the i-th frame. Note that such a UAV position is obtained from the state space updated by the action chosen in the DDPG structure.

In the j-th optimization iteration within the i-th frame, we initially set $m_{1,i}$ to the value from the previous iteration, denoted as $m_{1,i,j-1}$, which relies on the optimization results from the previous $j-1$-th iteration, to decouple the optimization variables. Then, we determine the UAV transmission power allocation factors $\{\rho_{n,i,j}\}$. After that, with the predetermined $\{\rho_{n,i,j}\}$, we find the optimal value of $m_{1,i,j}$ in the next step. Consequently, the obtained $\{\rho_{n,i,j}\}$ and $m_{1,i,j}$ are utilized in the $j+1$-th iteration.

3.3.1. Optimization of UAV Transmission Power Allocation Factors

During the j-th iteration, when we keep m_1 as a constant, it is evident that $m_2 = M - m_1$ is also unchanging. Consequently, our current goal is to attain the best power allocation factors $\{\rho_{n,i,j}\}$ at the UAV that minimize the maximum end-to-end decoding error rate among all GUEs. As a result, **P1** transforms into **P2** under the fixed value of m_1:

$$\begin{aligned}
\textbf{P2:}\ & \underset{\{\rho_{n,i,j}\}}{\textbf{Minimize}}\ \max_{\forall n\in\mathcal{N}}\{\epsilon_{n,i,j}^{\mathrm{tot}}\} \\
\textbf{s. t.}\ & \sum_{n=1}^{N}\rho_{n,i,j} = 1,
\end{aligned} \tag{27}$$

where $\{\rho_{n,i,j}\}$ represents the power allocation factors at the UAV, and $\epsilon^{\text{tot}}_{n,i,j}$ is the end-to-end decoding error probability of the n-th GUE in the j-th optimization iteration within the i-th frame.

P2 remains a challenging min-max optimization problem. To tackle this, we further break down **P2** into N sub-problems, and considering the n-th GUE, we construct:

$$\begin{aligned} \textbf{P2A:}\ \ & \underset{\{\rho_{\chi,i,j}\}}{\textbf{Minimize}}\ \epsilon^{\text{tot}}_{n,i,j} \\ \textbf{s. t.}\ \ & \sum_{\chi=1}^{N} \rho_{\chi,i,j} = 1, \\ & \epsilon^{\text{tot}}_{n,i,j} \geq \epsilon^{\text{tot}}_{k,i,j}, \quad \forall k \neq n \in \mathcal{N} \end{aligned} \tag{28}$$

We formulate a sub-problem **P2A** for each GUE $n \in \mathcal{N}$. In **P2A**, we focus on minimizing the end-to-end decoding error probability only for a single GUE. $\epsilon^{\text{tot}}_{n,i,j} \geq \epsilon^{\text{tot}}_{k,i,j}$, $\forall k \neq n \in \mathcal{N}$ ensures that this minimized error probability is the maximum among all GUEs, making the attained power allocation factors $\{\rho_{n,i,j}\}$ a potential solution for **P2**. To obtain the solution for **P2** from **P2A**, we introduce Lemma 1.

Lemma 1. *Among all the sub-problems* ***P2A****, the one that attains the minimum value in the objective function shares the same solution with* ***P2****.*

Proof. Assume that the t-th sub-problem is the one that attains the minimum value of the objective function, meaning $\epsilon^{\text{tot}*}_{t,i,j} < \epsilon^{\text{tot}*}_{v,i,j}$ for all GUEs $v \neq t \in \mathcal{N}$, and then when we apply the solution obtained from the t-th sub-problem to **P2**, the objective function's value must remain the same as $\epsilon^{\text{tot}*}_{t,i,j}$.

If the solution of **P2** differs from the solution of the t-th sub-problem, denoting $\epsilon^{\text{tot}*}_{u,i,j}, u \neq t$ as the minimum end-to-end decoding error probability based on the solution of **P2**, we would naturally expect that $\epsilon^{\text{tot}*}_{u,i,j} < \epsilon^{\text{tot}*}_{t,i,j}$ because **P2** is a minimization problem. This implies that the solution leading to $\epsilon^{\text{tot}*}_{u,i,j}$ in **P2** must be the solution of the u-th sub-problem. However, it contradicts our initial assumption that $\epsilon^{\text{tot}*}_{t,i,j} < \epsilon^{\text{tot}*}_{u,i,j}$, since $\epsilon^{\text{tot}*}_{u,i,j} < \epsilon^{\text{tot}*}_{t,i,j}$ should be satisfied. Thus, the solution of **P2** must be the same as that of the t-th sub-problem, which achieves the minimum value of the objective function among all the N sub-problems. □

When we aggregate the solutions obtained from all the N sub-problems, according to Lemma 1, we can confidently claim that the sub-problem solution that yields the lowest end-to-end decoding error probability in the objective function is the solution of **P2**.

Every sub-problem **P2A** can be addressed using a nonlinear optimization tool. However, it is worth noting that the Q function significantly escalates the computational complexity. To mitigate this challenge, following the approach in [21], we can approximate the Q function with the F function for any fixed m and D. For example, $Q(\gamma, m, D)$ can be approximated as $F^D_m(\gamma)$:

$$F^D_m(\gamma) = \begin{cases} 1, & \gamma \leq \theta^D_m \\ \frac{1}{2} - \alpha^D_m(\gamma - \beta^D_m), & \theta^D_m < \gamma < \kappa^D_m \\ 0, & \gamma \geq \kappa^D_m \end{cases} \tag{29}$$

where $\alpha^D_m = \sqrt{\frac{m}{2\pi 2^{\frac{2D}{m}} - 1}}$, $\beta^D_m = 2^{\frac{2D}{m}} - 1$, $\theta^D_m = \beta^D_m - \frac{1}{2\alpha^D_m}$ and $\kappa^D_m = \beta^D_m + \frac{1}{2\alpha^D_m}$.

Via (29), when m and D are fixed, the total end-to-end decoding error probability for the n-th GUE in the i-th frame is represented as $\epsilon^{\text{F}}_{n,i}$:

$$\begin{aligned}\epsilon^{\text{tot}}_{n,i} \approx \epsilon^{\text{F}}_{n,i} = &\sum_{c=1}^{C} X_{c,n,i}(1-Y_{c,i})F^{D_{\text{uav,i}}}_{m_{2,i}}(\gamma_{\text{uav,i}}) + \\ &\sum_{k=1}^{n-1} F^{D_{k,i}}_{m_{1,i}}(\gamma_{n,k,i}) + F^{D_{n,i}}_{m_{1,i}}(\gamma_{n,i}).\end{aligned} \tag{30}$$

We can subsequently convert **P2A** into following **P2B**:

$$\begin{aligned}\textbf{P2B:}\quad & \underset{\{\rho_{\chi,i,j}\}}{\textbf{Minimize}}\ \epsilon^{\text{F}}_{n,i,j} \\ \textbf{s. t.}\quad & \sum_{\chi=1}^{N} \rho_{\chi,i,j} = 1, \\ & \epsilon^{\text{F}}_{n,i,j} \geq \epsilon^{\text{F}}_{k,i,j}, \quad \forall k \neq n \in \mathcal{N} \\ & \theta^{D_{\chi,i,j}}_{m_{1,i,j}} < \gamma_{\chi,i,j} < \kappa^{D_{\chi,i,j}}_{m_{1,i,j}}, \quad \forall \chi \in \mathcal{N} \\ & \theta^{D_{v,i,j}}_{m_{1,i,j}} < \gamma_{\chi,v,i,j} < \kappa^{D_{v,i,j}}_{m_{1,i,j}}, \quad \forall v \leq \chi - 1, v \in \mathcal{N}\end{aligned} \tag{31}$$

where $\epsilon^{\text{F}}_{n,i,j}$ represents the value of $\epsilon^{\text{F}}_{n,i}$ in the j-th optimization iteration during the i-th frame. We can then solve **P2B** by using a nonlinear optimization tool without the inclusion of the Q function, resulting in a reduced computational complexity while sacrificing accuracy due to the approximation. The choice between solving **P2A** or **P2B** should be made considering the trade-off between solution accuracy and computational complexity.

By solving either **P2A** or **P2B**, we can determine the optimal power allocation factors $\{\rho_n\}^*$ on the UAV. These obtained $\{\rho_n\}^*$ in the j-th optimization iteration during the i-th frame are represented as $\{\rho_{n,i,j}\}$.

3.3.2. Optimization of the Length of DL Phase

During the second step of the two-step alternating optimization subroutine, we keep the power allocation factors on the UAV fixed as $\{\rho_{n,i,j}\}$, and this transforms **P1** into **P3** for determining the optimal duration of the DL phase in the j-th iteration during the i-th frame.

$$\begin{aligned}\textbf{P3:}\quad & \underset{m_{1,i,j}}{\textbf{Minimize}}\ \max_{\forall n \in \mathcal{N}} \{\epsilon^{\text{tot}}_{n,i,j}\} \\ \textbf{s. t.}\quad & m_{1,i,j} + m_{2,i,j} = M, \\ & m_{1,i,j}, m_{2,i,j} \in \mathbb{Z},\end{aligned} \tag{32}$$

where $m_{1,i,j}$ and $m_{2,i,j}$ represent the duration of the DL phase and requesting phase in the j-th optimization iteration during the i-th frame, respectively.

P3 is a discrete optimization problem, and when M is large, using exhaustive search becomes inefficient. To address this, we can initially treat $m_{1,i,j}$ as a continuous variable and solve **P3** without considering the integer limitation by using a nonlinear optimization tool. Similarly, as in **P2**, we can employ a similar approach to decompose **P3** into several minimization sub-problems. After completing the two-step alternating optimization subroutine, the optimal $m_{1,i}$ is determined by rounding the continuous solution to the nearest integer.

By iteratively solving **P2A**/**P2B** and **P3**, we can obtain the solution of **P1** once they converge. Algorithm 1 below outlines the details of the proposed two-step alternating optimization subroutine.

Algorithm 1 Two-step alternating optimization subroutine

```
Initialization:
1. Initialize $\{\rho_{n,i,0}\}$, $m_{1,i,0}$.
Actions:
1. For $j = 1 : \mathrm{J}_{\max}$
2. Obtain $\{\rho_{n,i,j}\}$ by solving P2A/P2B with $m_{1,i,j-1}$.
3. Obtain $m_{1,i,j}$ by solving P3 with $\{\rho_{n,i,j}\}$.
4. End If converged.
```

Note that in the last iteration, we must perform action 2 one more time to acquire the final power allocation factors $\{\rho_{n,i}\}$ on the UAV during the i-th frame.

3.4. Joint Trajectory Design and Resource Optimization Framework in the UAV-Assisted Network

In this section, we will explicitly illustrate the comprehensive framework in the considered UAV-assisted downlink network. The specific details can be found in Algorithm 2, presented below.

Algorithm 2 Framework in the UAV-assisted Network

```
Initialization:
1. Initialize caching size limitation $C_{\text{uav}}$ at the UAV, total length of a frame $M$,
   transmission power $P_{\text{BS}}$ from the BS to the UAV during the requesting phase, and
   maximum available transmission power $P_{\max}$ at the UAV during the DL phase.
2. Initialize all neural networks and the experience replay memory.
Actions:
1. Obtain initial state $s^0$.
2. For $i = 1 : I$
3. Check all the content requests from the GUEs with the cached contents at the UAV,
   and generate $\{X_{c,n,i}\}$, $\{Y_{c,i}\}$ and $\{Z_{c,i}\}$.
4. Determine the sampling rate selection and UAV movement action $a^i$ by the actor
   network according to current state $s^i$;
5. Obtain the location of UAV $(x_i^{\text{uav}}, y_i^{\text{uav}})$, which is given by the updated state space,
   and calculate the channel coefficients $h_{\text{uav,i}}$ and $\{h_{n,i}\}$.
6. Reorder the GUEs into an increasing order, i.e., $|h_{1,i}| \leq |h_{2,i}| \leq \ldots, \leq |h_{N,i}|$.
7. With given $h_{\text{uav,i}}$ and $\{h_{n,i}\}$, obtain the transmission power allocation factors $\{\rho_{n,i}\}$
   at the UAV, and the length of the DL phase in the $i$-th frame $m_{1,i}$ via Algorithm 1.
8. Observe reward $r^i$ and new state $s^{i+1}$. Update the caching list at the UAV, calculate
   the popularity of each content $\{O_{c,i}\}$, and then update the cache.
9. Store transition $(s^i, a^i, r^i, s^{i+1})$ in the experience replay memory;
10. Sample a random minibatch transition from the experience replay memory;
11. Train the critic and actor network, respectively;
12. Update target networks.
13. End for.
```

3.5. Convergence and Complexity Analysis

3.5.1. Convergence

In our proposed approach, we conduct training on the actor network $\mu(s|\psi^\mu)$ and critic network $Q(s,a|\psi^Q)$ using a gradient descent with exponentially decayed learning rates. Consequently, the weights ψ^μ and ψ^Q will reach convergence after a finite number of iterations, ensuring the overall convergence of the proposed algorithm. Although it is challenging to theoretically analyze the time required for the convergence prior to network training, we rely on simulations to demonstrate the convergence of our proposed algorithm, as indicated in the numerical results.

For the embedded two-step alternating optimization subroutine, we introduce the Lemma 2 below to analyze its convergence.

Lemma 2. *Algorithm 1 converges when the optimized objectives in* **P2** *and* **P3** *have the same value.*

Proof. Assume that the optimized objectives in **P2** and **P3** are $\epsilon_{p,i,j}^{\text{tot}*}$ and $\epsilon_{q,i,j}^{\text{tot}*}$, respectively. If $\epsilon_{p,i,j}^{\text{tot}*} \neq \epsilon_{q,i,j}^{\text{tot}*}$, then we have $\epsilon_{p,i,j}^{\text{tot}*} > \epsilon_{q,i,j}^{\text{tot}*}$, since the worst case in solving **P3** after **P2** is $\epsilon_{p,i,j}^{\text{tot}*} = \epsilon_{q,i,j}^{\text{tot}*}$ achieved by keeping $m_{1,i,j}$ to be the same as what was fixed in **P2**. This indicates that any improvement (i.e., reduction) in $\epsilon_{q,i,j}^{\text{tot}*}$ implies that we should have $\epsilon_{q,i,j}^{\text{tot}*} < \epsilon_{p,i,j}^{\text{tot}*}$. Consequently, if $\epsilon_{p,i,j}^{\text{tot}*} \neq \epsilon_{q,i,j}^{\text{tot}*}$, there must be an improvement in the optimized objective, and hence Algorithm 1 will continue until $\epsilon_{p,i,j}^{\text{tot}*} = \epsilon_{q,i,j}^{\text{tot}*}$. Therefore, Algorithm 1 will stop/converge when the optimized objectives in **P2** and **P3** have the same value. □

Furthermore, since both **P2** and **P3** are minimization problems, and there exists a lower bound on the total decoding error rate, together with the characterization in **Lemma 2**, the convergence of the embedded two-step alternating optimization subroutine is ensured.

3.5.2. Complexity

In our proposed algorithm, the well-trained actor network generates actions for the UAV during each frame. In our study, the computational complexity for the action generation for UAVs is expressed as $O(\sum_{l=1}^{l=N_h-1} N_l N_{l+1})$, where N_h is the number of network layers, and N_l represents the number of neurons in the l-th layer. Therefore, the total complexity of the DDPG algorithm is $O(I\sum_{l=1}^{l=N_h-1} N_l N_{l+1})$.

In the embedded two-step alternating optimization subroutine, **P2** and **P3** are solved iteratively based on Lemma 1. **P2B** involves N variables. Consequently, the necessary number of iterations is $O(\sqrt{N}\log_2(\frac{1}{\epsilon_0}))$, where ϵ_0 represents the desired accuracy of the interior-point method in solving **P2B**. Additionally, **P2B** encompasses at most $2N$ constraints, and hence the computational complexity for solving **P2B** is $O(\sqrt{N}\log_2(\frac{1}{\epsilon_0})(N+2N)^3)$, which is equivalent to $O(N^{3.5}\log_2(\frac{1}{\epsilon_0}))$. Therefore, during one iteration in Algorithm 1, the computational complexity for solving **P2** is $O(N^{4.5}\log_2(\frac{1}{\epsilon_0}))$, since we have N sub-problems. For each sub-problem of **P3**, we have one variable and one constraint when we relax m_1, and hence the computational complexity for solving **P3** is $O(N\log_2(\frac{1}{\epsilon_0}))$ during one iteration in Algorithm 1. Assuming that we have $\text{J}_{\max}$ iterations (the worst case) in Algorithm 1, the computational complexity is $O(\text{J}_{\max}(N^{4.5}\log_2(\frac{1}{\epsilon_0}) + N\log_2(\frac{1}{\epsilon_0})))$, which is equivalent to $O(\text{J}_{\max}N^{4.5}\log_2(\frac{1}{\epsilon_0}))$.

Finally, the overall computational complexity of Algorithm 2 is

$$O(I(\sum_{l=1}^{l=N_h-1} N_l N_{l+1} + \text{J}_{\max}N^{4.5}\log_2(\frac{1}{\epsilon_0}))). \tag{33}$$

4. Numerical Results

In this section, we present the numerical analysis of the minimized average maximum end-to-end decoding error probability among all GUEs by using the proposed algorithm. In our numerical results, we represent the end-to-end decoding error probability on a logarithmic scale. We first show the average min-max end-to-end decoding error rate versus the maximum block length constraint under different UAV transmission power constraints $P_{\max}$ as well as different UAV trajectories. Additionally, we explore the influence of cache size limitations and the length of the caching list at the UAV. Furthermore, we display the UAV trajectory optimized through the proposed two-step alternating optimization scheme embedded within the DDPG algorithm. Finally, we provide insights into the convergence performance of our proposed learning structure.

In the simulations, the channels are modeled as follows: For GUE n in the set $\mathcal{N}$, the channel is generated using the formula $h_n = \sqrt{\xi_0 d_n^{-\alpha_n}}\widetilde{g}_n$, where d_n represents the distance between the UAV and the n-th GUE, α_n is the path loss exponent, and $\widetilde{g}_n$ is the complex Gaussian distributed fading component for the n-th UE. In a similar manner, the channel between the BS and the UAV is characterized by $h_{\text{uav}} = \sqrt{\xi_0 d_{\text{uav}}^{-\alpha_{\text{uav}}}}\widetilde{g}_{\text{uav}}$, where d_{uav} denotes the distance from the BS to the UAV, α_{uav} is the path loss exponent for that link, and $\widetilde{g}_{\text{uav}}$ represents the complex Gaussian distributed fading component associated with the BS-UAV connection. Unless stated otherwise, the UAV serves 3 GUEs in the considered network. The simulation parameters are listed in Table 4, below.

Table 4. Summary of parameters.

Parameter	Definition	Value
α_{uav}	Path loss exponent from the BS to the UAV	2
α_n	Path loss exponent from the UAV to the n-th GUE	3.5
ξ_0	Path loss at the reference point $d_0 = 1$ m	−30 dB
σ^2	Noise power	−95 dBm
δ	Target update ratio	0.005
ζ	Discount factor	0.85
N_{b}	Minibatch size	64

In Figure 5, we analyze the average min-max error probability attained with the proposed algorithm considering three different UAV trajectories. In this figure, the curves in red and blue denote for the average min-max error rates under different UAV transmission power budgets $P_{\max}$, when the UAV is following a circular trajectory, while the dotted curve in black represents the average min-max error rate when the UAV has the optimal trajectory with $P_{\max} = 2W$, and the dash dotted curve in purple denotes the average min-max error probability when the UAV has a point-to-point (P2P) trajectory with $P_{\max} = 2W$. Such a P2P trajectory starts at (0, 380) and ends at (350, 100) with a straight line in the considered 400 m × 400 m square area. It is evident that UAV using the optimized trajectory, which is obtained from our proposed DDPG algorithm embedded with the two-step alternating optimization subroutine, results in the smallest average min-max error rate. We further observe that the min-max error probability is reduced when the blocklength constraint M increases, which is expected since increasing M is the same as extending the transmission time, resulting in less strict requirements on the coding rate. We further observe that enlarging the transmission power budget $P_{\max}$ at the UAV improves the performance as well. By increasing $P_{\max}$, we can obtain a higher SNR/SINR and hence improve the min-max error probability.

Next, we investigate the impact of the cache size limitation and the length of the caching list in Figure 6. This figure demonstrates the curves of the average min-max error probability versus the UAV's cache size limitation C_{uav}, where the red and blue curves represent the average min-max error rates for different lengths L of the caching list when the UAV follows a circling trajectory, and the dotted black line illustrates the average min-max error rate when the UAV adopts the optimal trajectory with $L = 100$. In Figure 6, it is readily observed that the performance with a larger caching list is consistently better than the one with a smaller caching list. This is due to the fact that a larger caching list increases the probability of caching all the popular contents, making the caching procedure more efficient. We additionally observe that a larger cache size results in a lower average min-max error rate. This is attributed to the improved caching capability at the UAV, allowing more contents to be served without consulting the BS and leading to an improved average min-max error probability. However, it is important to note that the improvement becomes smaller as the cache size is increased. This phenomenon occurs because the most popular content is the first to be cached, and further increasing the cache size mainly enables the UAV to store less popular contents, providing limited improvements in the min-max error rate. Last but not least, even when all contents are cached at the UAV, decoding errors may still occur during the DL phase. Consequently, the min-max end-to-end decoding error probability does not vanish by merely increasing the UAV's cache size.

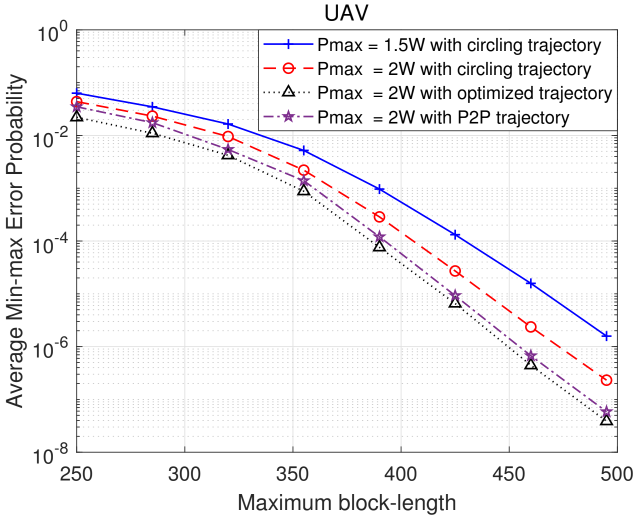


Figure 5. Influence of blocklength constraint.

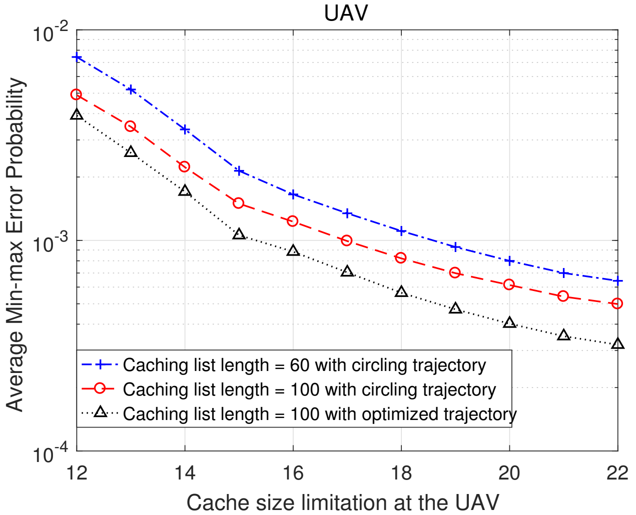


Figure 6. Influence of cache size limitation.

Furthermore, considering both the UAV trajectory design, transmission power allocations, and determination of the duration of the DL phase, we illustrate the optimal UAV trajectory in Figure 7. In the figure, the GUEs are randomly distributed within a circle whose center is located at (350, 100). The initial position of UAV is set to be (0, 380) and the BS is located at (350, 380). We can observe that the UAV will first fly somewhat towards the BS, since at that stage the error rate arising from the UAV downloading in the requesting phase has more impact on the end-to-end error probability. Afterwards, the UAV flies more towards the GUE cluster, since, at that stage, the decoding error rate in the DL phase dominates the overall error probability. We also observe that the UAV will continue hovering in a very small range close to the GUE cluster, which is due to the fact that such a position is the optimal location with which the overall end-to-end error rate is minimized. It is worth noting that when we alter the location of the GUE cluster, the UAV trajectory changes correspondingly.

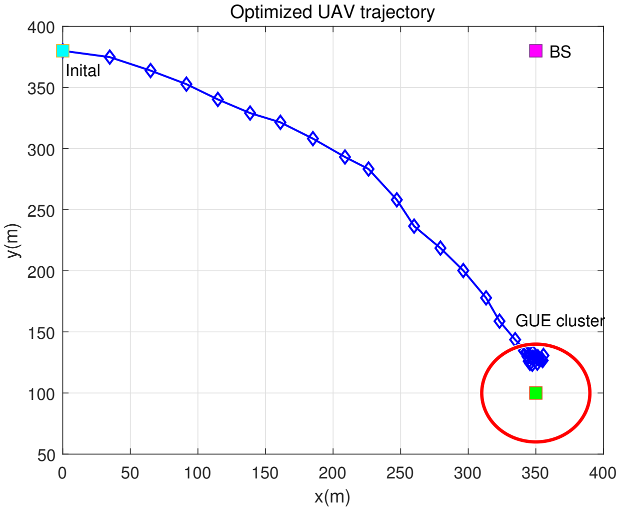


Figure 7. Optimized UAV trajectory.

Additionally, we evaluate the convergence of the proposed DDPG algorithm embedded with the two-step alternating optimization subroutine, as presented in Figure 8. This figure shows the reward curve as the number of training episodes grows. In Figure 8, the results reveal that the UAV trajectory training will converge around 710 episodes, which assures the feasibility of our proposed algorithm in addressing the global optimization problem.

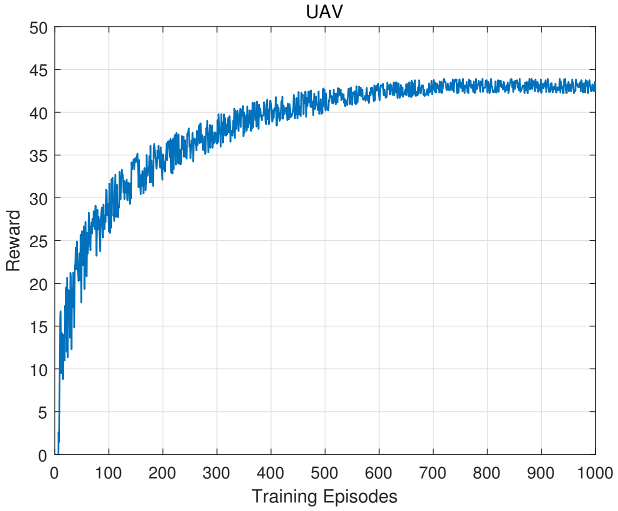


Figure 8. Convergence.

In Figure 9, we validate the effectiveness of hybrid frequency-division multiple access (FDMA)-NOMA to overcome the interference bottleneck as the number of GUEs grows. In the hybrid FDMA-NOMA, we first divide all the GUEs into different groups, and then FDMA is utilized between different groups and the GUEs in the same group receive data from the UAV via NOMA transmissions. To perform the hybrid FDMA-NOMA, we can simply execute step (6) for every group independently in Algorithm 2 and the rest of the steps remains the same. In Figure 9, we explicitly demonstrate the average min-max error probability under different grouping scenarios considering 4 and 6 GUEs. We observe that with the increasing number of GUEs, the average min-max error probability increases. We further notice that for the same number of GUEs, the more groups we have, the smaller average min-max error probability we can achieve. This is mainly due to the fact that with less GUEs in one NOMA group, the GUEs experience less impact from the SIC errors.

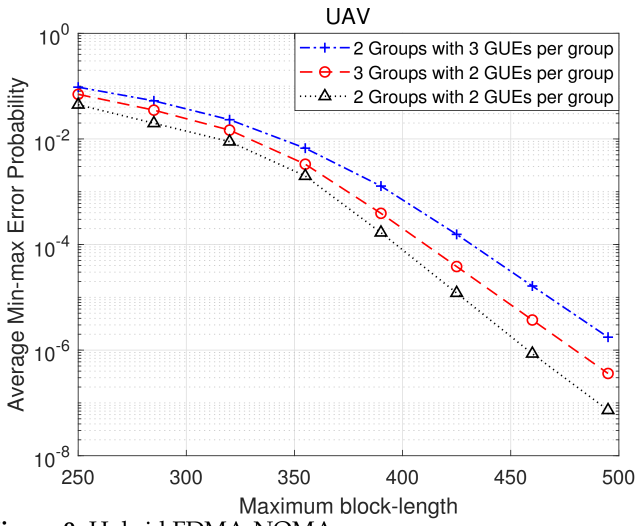


Figure 9. Hybrid FDMA-NOMA.

Finally, we plot Figure 10 to illustrate the optimized values of m_1 and m_2, as the maximum blocklength M keeps increasing. In order to focus on m_1 and m_2, we keep the UAV hovering within a small range of the final position obtained by the optimized UAV trajectory, as shown in Figure 7. In Figure 10, we continuously increase the maximum blocklength $M = m_1 + m_2$ to plot the average m_1 and m_2. We observe that with the increase in M, more time is allocated to the DL phase, resulting in a relatively fast growth in m_1. Compared with m_1, m_2 increases slowly. This can be attributed to the fact that during the DL phase, the UAV performs downlink NOMA transmissions to the GUEs, and allocating more time to such a phase mitigates the impacts from both the self-decoding error and SIC errors. Hence, the optimized allocation indicates that we benefit more from increasing m_1 rather than m_2, especially in later stages of the UAV flight, at which time most popular content is highly likely to be already stored at the UAV and communication with the BS has a lower priority.

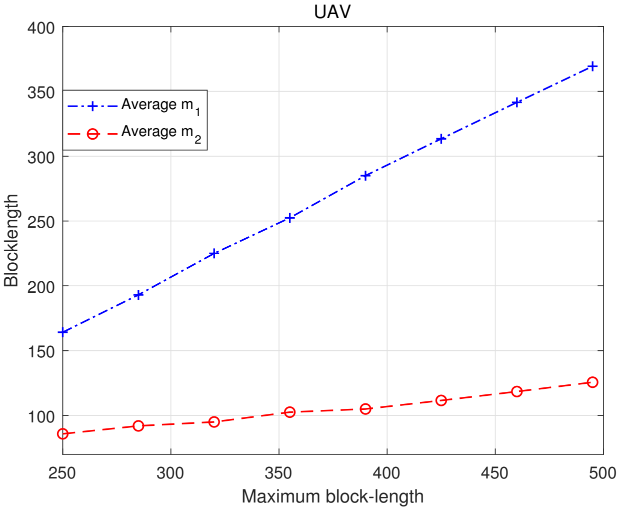


Figure 10. m_1 and m_2.

5. Conclusions

In this paper, we have investigated the reliability of a UAV-assisted downlink network with content caching and NOMA transmission in the FBL regime. We have first presented the system model and conducted an analysis of the FBL regime as well as the SINR when the NOMA transmission is employed. We then have addressed the end-to-end decoding error probability and introduced a caching policy for the UAV. We have subsequently formulated an optimization problem aimed at minimizing the average maximum end-to-end decoding error rate for all GUEs within specified time frames, subject to coding length and maximum UAV transmission power constraints. To address this problem, we have initially presented a DDPG learning-based approach to optimize the UAV trajectory. Furthermore, we have proposed a two-step alternating optimization subroutine to determine the optimal solutions of the transmission power allocation at the UAV and the duration of the DL phase for any given UAV position. Our numerical results indicate that the higher UAV power budget $P_{\max}$ results in a lower end-to-end decoding error rate, and increasing the maximum blocklength M enhances the network performance. We have also observed that content caching at the UAV significantly improves the end-to-end decoding error probability. Moreover, our optimized UAV trajectory consistently outperforms a circular trajectory in terms of the average min-max error probability. Furthermore, we have explicitly demonstrated the optimized UAV trajectory and the convergence performance to demonstrate the effectiveness of our proposed algorithm in this paper. We have additionally validated the effectiveness of hybrid FDMA-NOMA to overcome the interference bottleneck as the number of GUEs grows. Finally, we have concluded that it is more effective to allocate more time to the DL phase rather than the requesting phase, especially when the blocklength constraint is relaxed. Future work includes the investigation of the network with multiple UAVs, and with mobile GUEs to capture more challenging practical scenarios.

Author Contributions: Methodology, Y.Y.; Validation, Y.Y.; Formal analysis, Y.Y.; Investigation, Y.Y. and M.C.G.; Writing—original draft, Y.Y.; Writing—review & editing, M.C.G.; Supervision, M.C.G.; Project administration, Y.Y. All authors have read and agreed to the published version of the manuscript.

Funding: The work in this paper is supported by National Science Foundation Grant CNS 2221875.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Zhang, X.; Duan, L. Fast deployment of UAV networks for optimal wireless coverage. *IEEE Trans. Mob. Comput.* **2018**, *18*, 588–601. [CrossRef]
2. Mohammed, F.; Idries, A.; Mohamed, N.; Al-Jaroodi, J.; Jawhar, I. UAVs for smart cities: Opportunities and challenges. In Proceedings of the 2014 International Conference on Unmanned Aircraft Systems (ICUAS), Orlando, FL, USA, 27–30 May 2014; pp. 267–273.
3. Ullah, Z.; Al-Turjman, F.; Mostarda, L. Cognition in UAV-aided 5G and beyond communications: A survey. *IEEE Trans. Cogn. Commun. Netw.* **2020**, *6*, 872–891. [CrossRef]
4. Zeng, Y.; Zhang, R.; Lim, T.J. Wireless communications with unmanned aerial vehicles: Opportunities and challenges. *IEEE Commun. Mag.* **2016**, *54*, 36–42. [CrossRef]
5. Zhang, T.; Wang, Y.; Liu, Y.; Xu, W.; Nallanathan, A. Cache-enabling UAV communications: Network deployment and resource allocation. *IEEE Trans. Wirel. Commun.* **2020**, *19*, 7470–7483. [CrossRef]
6. Lv, L.; Chen, J.; Ni, Q.; Ding, Z.; Jiang, H. Cognitive non-orthogonal multiple access with cooperative relaying: A new wireless frontier for 5G spectrum sharing. *IEEE Commun. Mag.* **2018**, *56*, 188–195. [CrossRef]
7. Yu, Y.; Chen, H.; Li, Y.; Ding, Z.; Vucetic, B. On the performance of non-orthogonal multiple access in short-packet communications. *IEEE Commun. Lett.* **2017**, *22*, 590–593. [CrossRef]
8. Liu, X.; Wang, J.; Zhao, N.; Chen, Y.; Zhang, S.; Ding, Z.; Yu, F.R. Placement and power allocation for NOMA-UAV networks. *IEEE Wirel. Commun. Lett.* **2019**, *8*, 965–968. [CrossRef]
9. Sachs, J.; Wikstrom, G.; Dudda, T.; Baldemair, R.; Kittichokechai, K. 5G radio network design for ultra-reliable low-latency communication. *IEEE Netw.* **2018**, *32*, 24–31. [CrossRef]
10. Polyanskiy, Y.; Poor, H.V.; Verdú, S. Channel coding rate in the finite blocklength regime. *IEEE Trans. Inf. Theory* **2010**, *56*, 2307–2359. [CrossRef]
11. Sun, C.; She, C.; Yang, C.; Quek, T.Q.; Li, Y.; Vucetic, B. Optimizing resource allocation in the short blocklength regime for ultra-reliable and low-latency communications. *IEEE Trans. Wirel. Commun.* **2018**, *18*, 402–415. [CrossRef]
12. Thanh, P.D.; Giang, H.T.H.; Koo, I. UAV-assisted NOMA downlink communications based on content caching. In Proceedings of the 2020 International Conference on Information and Communication Technology Convergence (ICTC), Jeju, Korea, 21–23 October 2020; pp. 786–791.
13. Luo, J.; Song, J.; Zheng, F.-C.; Gao, L.; Wang, T. User-centric UAV deployment and content placement in cache-enabled multi-UAV networks. *IEEE Trans. Veh. Technol.* **2022**, *71*, 5656–5660. [CrossRef]

14. Amjad, M.; Musavian, L.; Aissa, S. NOMA versus OMA in finite blocklength regime: Link-layer rate performance. *IEEE Trans. Veh. Technol.* **2020**, *69*, 16253–16257. [CrossRef]
15. Singh, S.K.; Agrawal, K.; Singh, K.; Chen, Y.-M.; Li, C.-P. Performance Analysis and Optimization of RSMA Enabled UAV-Aided IBL and FBL Communication with Imperfect SIC and CSI. *IEEE Trans. Wirel. Commun.* **2022**, *22*, 3714–3732. [CrossRef]
16. Wang, Y.; Zhou, X.; Zhuang, Z.; Sun, L.; Qian, Y.; Lu, J.; Shu, F. UAV-Enabled Secure Communication With Finite Blocklength. *IEEE Trans. Veh. Technol.* **2020**, *69*, 16309–16313. [CrossRef]
17. Pan, C.; Ren, H.; Deng, Y.; Elkashlan, M.; Nallanathan, A. Joint Blocklength and Location Optimization for URLLC-Enabled UAV Relay Systems. *IEEE Commun. Lett.* **2019**, *23*, 498–501. [CrossRef]
18. Chen, K.; Wang, Y.; Zhao, J.; Wang, X.; Fei, Z. URLLC-Oriented Joint Power Control and Resource Allocation in UAV-Assisted Networks. *IEEE Commun. Lett.* **2021**, *8*, 10103–10116. [CrossRef]
19. Yang, Y.; Gursoy, M.C. Reliability-Oriented Designs in UAV-assisted NOMA Transmission with Finite Blocklength Codes and Content Caching. In Proceedings of the 2023 32nd International Conference on Computer Communications and Networks (ICCCN), Honolulu, HI, USA, 24–27 July 2023; pp. 1–8.
20. Lillicrap, T.P.; Hunt, J.J.; Pritzel, A.; Heess, N.; Erez, T.; Tassa, Y.; Silver, D.; Wierstra, D. Continuous control with deep reinforcement learning. *arXiv* **2015**, arXiv:1509.02971.
21. Makki, B.; Svensson, T.; Zorzi, M. Finite block-length analysis of the incremental redundancy HARQ. *IEEE Wirel. Commun. Lett.* **2014**, *3*, 529–532. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.