Game Theory Empowered Carbon-Intelligent Federated Multi-Edge Caching for Industrial Internet of Things

Jie Liang, Zhengxin Yu, Haris Pervaiz, Guhan Zheng, and Neeraj Suri

***Abstract*—To navigate the carbon emission and functional challenges associated with edge caching within heterogeneous Industrial Internet of Things (IIoT) spanning energy use, cache hit rate, and bandwidth usage, this paper proposes a novel Game Theory Empowered Carbon-Intelligent Federated Multi-Edge Caching framework (GT-FMC). The proposed framework enables distributed collaborative caching by intelligently coordinating edge nodes to optimize content decisions while efficiently integrating content providers (CPs), edge nodes, and users with energy-aware strategies. In GT-FMC, a lightweight federated content popularity prediction method based on Temporal Convolutional Networks (TCN) is introduced to collaboratively learn global content popularity while reducing prediction energy cost. The energy-aware utilities of the three involved parties are jointly formulated as a coupled non-linear optimization problem. To address this challenge, a two-stage game-theoretic algorithm is designed. Experimental results on a real-world testbed show that GT-FMC achieves up to 77.9% of Oracle in cache hit rate and 10.6%–32.4% reduction in transmission energy consumption compared to baseline methods. Complementary evaluations also validate the game-theoretic design's effectiveness.**

***Index Terms*—multi-edge collaborative caching, federated deep learning, game theory, content popularity prediction.**

I. Introduction

THE Industrial Internet of Things (IIoT) is transforming industries by enabling real-time data processing, automation, and intelligent decision-making [1], [2]. It integrates advanced sensors and devices to enhance industrial processes. However, as IIoT systems scale, the volume of data and associated energy demands grow rapidly, leading to significant carbon emissions. According to recent studies, global data infrastructure—including networking and storage—accounts for over 2% of total CO_2 emissions annually. Traditional centralized cloud caching exacerbates this issue due to long-distance data transmission and high energy use in data centers. In contrast, edge caching can improve IIoT efficiency by storing frequently accessed data closer to end users, reducing latency, bandwidth usage, and data transmission energy [3], [4]. In edge caching, caching user-preferred content at edge nodes (*e.g.*, base station, roadside unit, and smart gateways) can minimize the need for continuous data transmission, leading to enhanced energy savings [5], [6], [7].

The complex multi-node environment of IIoT introduces significant complexity to edge caching schemes due to the traditionally decoupled nature of caching mechanisms [8]. In IIoT systems, both edge nodes and users (e.g., sensors, smart industrial devices, and AR/VR terminals) are integral components of the industrial infrastructure, while Content Providers (CPs) typically reside in the cloud data center, functioning as centralized sources of content and intelligence. CPs provide content recommendations by analysing users' requests, constrained by privacy concerns. CPs aim to maximize content dissemination revenue. Edge nodes cache the content recommended by CPs and enable users to access the content. They focus on efficient resource utilization and energy efficiency. Furthermore, distributed users generate vast amounts of data through their devices and sensors. CPs utilize this data to predict content popularity. The cache hit rate of content recommended by CPs directly influences user satisfaction and engagement, ultimately driving providers' profitability. Users seek instant access to diverse content with minimal latency and seamless delivery within optimal topology between users and edge nodes. Thus, in IIoT, CPs, edge nodes, and users are tightly coupled, with caching decisions directly affecting both performance and sustainability. However, these three parties also have different and often conflicting priorities. For example, CPs push for extensive content replication to enhance reach, which may overwhelm the limited storage and computational resources of edge nodes. Users' demand for instant access to diverse content can conflict with both CPs' and edge nodes' goals. Therefore, deciding which content to cache across edge nodes is difficult, as requires balancing competing interests. However, achieving this balance is challenging due to dynamic user demands, limited edge resources, and the need for efficient coordination.

As the number of IIoT users grows, the volume of discretely distributed data increases, resulting in higher computational

This work was partly supported by the National Natural Science Foundation of China under Grant No. 62202103, the Central Funds Guiding the Local Science and Technology Development under Grant No. 2022L3004, the Fujian Province Technology and Economy Integration Service Platform under Grant No. 2023XRH001, the Fuzhou-Xiamen-Quanzhou National Independent Innovation Demonstration Zone Collaborative Innovation Platform under Grant No. 2022FX5, and the Collaborative Innovation Platform Project of Fuzhou City under Grant No. 2023-P-002. *(Corresponding author: Guhan Zheng)*

Jie Liang is with the College of Computer and Data Science, Fuzhou University, Fuzhou 350116, China, the Engineering Research Center of Big Data Intelligence, Ministry of Education, Fuzhou 350002, China, and also with the Fujian Key Laboratory of Network Computing and Intelligent Information Processing (Fuzhou University), Fuzhou 350116, China. (e-mail: steveliangjie@163.com)

Zhengxin Yu, Guhan Zheng, Neeraj Suri are with the School of Computing and Communications, Lancaster University, Lancaster LA1 4YW, United Kingdom. (e-mail: z.yu8@lancaster.ac.uk; guhan.zheng@ieee.org; neeraj.suri@lancaster.ac.uk)

Haris Pervaiz is with the School of Computer Science and Electronic Engineering (CSEE), University of Essex, CO4 3SQ, United Kingdom. (Email: haris.pervaiz@essex.ac.uk).

and communication overhead for single-node edge caching. However, traditional centralized training frameworks struggle with scalability, limiting their effectiveness in large-scale deployments [9]. To overcome these challenges, Federated Learning (FL) based caching schemes [10], [11], [12] have been proposed to enhance edge caching efficiency by enabling distributed model training across edge devices without sharing raw data. This approach preserves user privacy, reduces communication overhead, and enables seamless scalability across diverse users. By leveraging local content access logs and context-specific request patterns collected from IIoT users, each edge node trains a local content popularity prediction model. The model aggregation is then performed by a cloud-based central controller. By leveraging local content request data, FL trains a shared content popularity prediction model, with model aggregation performed at a central node. This decentralized paradigm accelerates model convergence, allowing edge nodes to optimize caching strategies based on content popularity.

FL-based edge caching faces significant system heterogeneity, as edge nodes in IIoT environments vary in computing power, energy availability, and network conditions. However, existing FL-based caching schemes neglect system heterogeneity [13] and user-edge node coupling by randomly selecting users as federated nodes, leading to inefficiencies in model training and resource allocation. Establishing a stable user-to-edge node topology remains a critical challenge, as uneven user distribution causes system bottlenecks [14]. Moreover, in distributed caching, the efficient collaboration among edge nodes is often hindered by improper resource configuration and the complexity of capturing relationships between discrete user features and diverse content libraries. Furthermore, the limited computational power and storage capacity of edge devices can lead to inefficient energy allocation, complicating the integration of carbon-intelligent strategies. Although edge caching can reduce energy consumption to a certain extent, existing frameworks often lack carbon-intelligent designs that can minimize carbon emissions.

To address above these challenges, we propose a Game Theory Empowered Carbon-Intelligent Federated Multi-edge Caching Framework (GT-FMC) to optimize caching decisions among edge nodes while intelligent reducing carbon emissions. A lightweight Temporal Convolutional Network (TCN) is used to predict content popularity in order to optimize caching decisions by leveraging parallel computation and low memory consumption to minimize latency and energy usage while enhancing prediction accuracy. Moreover, a two-stage game-theoretic algorithm is designed to resolve conflicts of utilities amongst CPs, edge nodes, and users in heterogeneous systems ensuring a balanced trade-off between content dissemination, energy utilization, and low-latency access.

The main contributions are as follows:

- We propose a distributed collaborative caching framework that reduces carbon emissions by intelligently coordinating edge nodes to optimize content decisions while optimally integrating content providers, edge nodes, and users. This proposed framework involves content recommendation, caching strategy, and user-to-edge node topology, enabling a comprehensive evaluation of a carbon-intelligent paradigm.
- We develop a lightweight federated content popularity prediction method that integrates a Temporal Convolutional Network to collaboratively learn global content popularity by capturing temporal patterns in user content requests. To further enhance efficiency, we incorporate an effective model compression technique, which significantly reduces communication overhead while maintaining high prediction accuracy, leading to reduced energy consumption.
- We investigate the joint utility optimization of carbon-aware three parties in heterogeneous distributed caching systems, considering the topology relationship, recommendation strategy, and caching strategy. Facing a three-party coupled intricate non-linear optimization problem, a two-stage game-theoretic algorithm is presented. At each stage, we decouple the three-party coupled problem into a tractable two-party interaction and design game mechanisms based on the matching game and the Stackelberg game, respectively. Through iterative optimization across both stages, our approach seeks to maximize the benefits of all three parties while minimizing carbon emissions associated with energy consumption.

The rest of this paper is organized as follows: Section II provides the related works. In Section III, we present the system model and formulate the carbon-aware optimization problem. Section IV detail presents the caching mechanism and the two-stage game-theoretic algorithm. In Section V, the performance evaluation is conducted through experiments. Finally, Section VI concludes the paper and discusses its potential applications in IIoT scenarios.

II. Related Works

With the rapid increase in network traffic, traditional caching strategies such as First In First Out (FIFO), Least Frequently Used (LFU), and Least Recently Used (LRU) rely on static rules for updating cached content, which limits their effectiveness in complex multi-edge environments. In the context of multi-edge caching, it is essential to comprehensively model and consider all system components, including content providers, edge nodes, and users, to design optimized carbon intelligent collaborative caching solutions. This section analyses the related studies from the perspectives of collaborative caching and FL-based edge caching.

A. Collaborative Caching

Qin *et al.* [15] proposed a NOMA-enabled multi-UAV collaborative caching network, where UAVs cache and share popular content, optimizing caching decisions, 3D trajectory planning, and resource allocation to minimize content retrieval delay. Lin *et al.* [5] introduced a user preference-aware content caching and migration scheme for video content delivery in edge networks, leveraging collective reinforcement learning and Lyapunov optimization to efficiently balance performance and cost. Zong *et al.* [6] designed an ensemble learning-based edge caching strategy, combining classic LFU/LRU with

LSTM-based time-series analysis, aiming to optimize cache configurations for heterogeneous edge nodes. Liu *et al.* [16] proposed a UAV-assisted edge caching network that optimizes cache placement, user association, and trajectory planning using a two-timescale deep reinforcement learning algorithm to maximize throughput. Most existing approaches to edge caching focus on the collaborative optimization between users and edge nodes, often neglecting the cooperation and game-theoretic interactions between service/content providers and edge nodes within IIoT multi-edge caching frameworks.

Tsigkari *et al.* [8] proposed a network-economic framework for cache-aware recommendations, leveraging Nash bargaining to align the financial objectives of content providers and caching networks, ensuring fair and Pareto-optimal cooperation. Fu *et al.* [17] explored the interplay between personalized bundle recommendation and cache decisions in wireless edge caching networks, formulating a revenue-maximization problem that considers user-specific recommendation quality, recommendation volume, and cache capacity constraints. Sun *et al.* [18] proposed a recommendation-enabled edge caching framework in mobile two-tier computing networks, integrating recommender systems with edge caching to enhance direct and soft hits, thereby improving resource utilization.

While these studies introduce novel schemes for cooperation and game theory between content providers and edge nodes, they overlook the critical impact of user-to-edge node topology on the system's utility, often substituting this dynamic with a static topology. In contrast, we propose a comprehensive distributed collaborative caching framework involving content providers, edge nodes, and users. Special emphasis is placed on modeling the intricate interactions in collaborative caching, ensuring a more accurate and realistic representation of multi-edge caching systems in practical environments.

B. Federated Learning based Edge Caching

Zhang *et al.* [19] proposed a federated learning-based caching and offloading resource allocation strategy in vehicular social networks, which leverages mobile social similarity for collaborative resource allocation. Their approach employs optimization algorithms to minimize task processing delay while achieving superior network load balancing. Chen *et al.* [20] addressed the challenge of dynamic content caching in distributed autonomous networks by proposing a contextual bandit-based algorithm combined with federated learning for collaborative edge caching. Wu *et al.* [21] introduced a cooperative caching scheme in vehicular edge networks, based on asynchronous federated deep reinforcement learning, aimed at predicting popular content and identifying optimal caching locations for these predicted contents. Chen *et al.* [11] tackled the issue of inefficient collaboration and adversarial threats in multi-edge caching by proposing a federated deep learning-based framework with multi-dimensional cache partitioning and proactive cache replacement. Zhou *et al.* [12] developed a content recommendation-based edge caching method in multi-tier edge-cloud networks, leveraging a federated distributed DDPG-based approach to minimize content delivery delay and maximize cache hit rate.

In general, existing studies on FL-based edge caching have demonstrated its potential in improving model scalability, cache allocations, and collaborative recommendation. However, they overlook the long-term and short-term variations in content popularity during dynamic user request processes, while the communication overheads introduced by complex system models in FL cannot be ignored. Therefore, effectively integrating FL into collaborative caching frameworks remains a challenging task.

III. System Model and Problem Formulation

This section presents the distributed caching system, detailing the caching strategy, topology, cache request, content popularity, and content recommendation model employed in the caching process. Furthermore, we analysis the utility models of users, edge nodes, and the CP, formulating their carbon-aware utility optimization problems.

A. System Description

We consider a hierarchical IIoT system consisting of J edge nodes, collectively represented by the set $\mathcal{C}$. Each edge node c_j $(j = 1, \ldots, J)$ is equipped with a caching capacity denoted by s_j. Furthermore, users are spatially distributed within the service region following a Poisson Point Process (PPP) [22] and are represented by the set $\mathcal{U}$. Each user u_h $(h = 1, \ldots, H)$ requests content from the edge node that minimizes their energy consumption. If the requested content is unavailable in any edge node's cache, it is served directly from the CP, which located at the cloud data center and maintains the complete content catalog $\mathcal{K}$.

1) Edge Caching Strategy: The caching strategy across all edge nodes is represented by a binary decision matrix:

$$X = \{x_{ij} \mid x_{ij} \in \{0,1\}, i \in \mathcal{K}, j = 1, \ldots, J\}, \quad (1)$$

where $x_{ij} = 1$ indicates that content i is cached in the caching space of edge node c_j. In addition, δ_i denotes the size of content i (in GB). The caching space constraint is given by:

$$\sum_{i \in \mathcal{K}} \delta_i x_{ij} \leq s_j, \quad \forall j = 1, \ldots, J. \quad (2)$$

2) Topology of the Edge Caching Framework: The CP is connected to the edge nodes via wired backhaul links. Each user u_h is associated with a primary edge node but can also access collaborate nodes through routing. The user-to-edge node topology is represented by the following matrix:

$$K = \{k_{hj} \mid k_{hj} \in \{0,1\}, h = 1, \ldots, H, j = 1, \ldots, J\}, \quad (3)$$

where $k_{hj} = 1$ indicates that user u_h is directly connected to edge node c_j; otherwise, $k_{hj} = 0$. The connection constraint is given by:

$$\sum_{u_h \in \mathcal{U}} k_{hj} = 1, \quad \forall j = 1, \ldots, J. \quad (4)$$

In the network topology, the distances between users and edge nodes are denoted by D (measured in km), defined as:

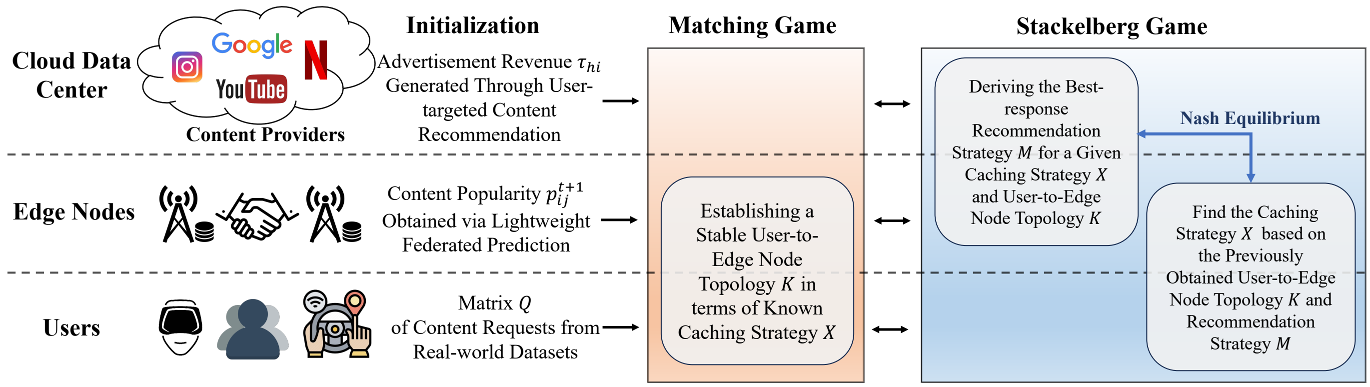

Fig. 1. The proposed Game Theory Empowered Federated Multi-Edge Caching Framework.

TABLE I
NOTATION TABLE

Notation	Description
$\mathcal{U}$	Set of users
$\mathcal{C}$	Set of edge nodes
$\mathcal{K}$	Set of contents
k_{hj}	Topology indicator between user u_h and edge node c_j
d_{hj}	Distance between user u_h and edge node c_j
$g_{jj'}$	Distance between edge nodes c_j and $c_{j'}$
$\hat{D}$, $\hat{G}$	Normalized user-to-edge and edge-to-edge distances
$\mathcal{C}(u_h)$	Set of edge nodes accessible to user u_h
x_{ij}	Caching decision for content i at edge node c_j
m_{hi}	Recommendation decision for content i to user u_h
q_{hi}	Request indicator of user u_h for content i
s_j	Cache capacity of edge node c_j
δ_i	Size of content i
N_h	Number of contents recommended to user u_h
p_{ij}^t	Local popularity of content i at node j at time t
$\bar{p}_{ij}^{t+1}$	Predicted popularity of content i at node j for time $t+1$
ω_j	Local TCN model parameters at edge node c_j
$\hat{\omega}_j$	Compressed parameters of ω_j using 3LC
$\mathbf{r}_{\text{ZR}}$	Zero-run encoded representation of compressed weights
ν	Sparsity control parameter in 3LC
ρ	Recommendation discount factor in CP revenue
l	Historical time window length
η	Long-short term balance factor
γ_s	Satisfaction-price refraction coefficient
γ_e	Energy-price refraction coefficient
α	Path loss exponent
λ_{CP}	Content provider transmission cost
λ_{MEC}	Edge node transmission cost
P_{wireless}	Wireless transmission power consumption
P_{wired}	Wired transmission power consumption
r_{CP}	Content provider transmission rate
r_j	Edge node transmission rate
$\epsilon_{\tilde{M}}, \epsilon_{\tilde{X}}$	Error tolerance in continuous relaxation constraints
U^{user}	Utility of users
U^{CP}	Utility of the content provider
U^{MEC}	Utility of the edge nodes
$\tilde{X}$, $\tilde{M}$	Relaxed continuous caching and recommendation strategies
W	Learnable weight matrix in caching strategy embedding
$\tilde{\Phi}(\cdot)$	Sigmoid-like function

$$D = \{d_{hj} \mid h = 1, \ldots, H, j = 1, \ldots, J\}. \tag{5}$$

Similarly, the distances between edge nodes are represented by G, where:

$$G = \{g_{jj'} \mid j, j' = 1, \ldots, J\}. \tag{6}$$

The normalized distance matrices are denoted by $\hat{D}$ and $\hat{G}$, respectively.

For a given user u_h, the set of accessible caching spaces $\mathcal{C}(u_h)$ is determined based on the user-to-edge node topology matrix K. The construction of $\mathcal{C}(u_h)$ involves two steps: initially, all edge nodes are sorted by their physical distance d_{hj} to the user u_h. Then, the edge node directly connected to the user is promoted to the first position in the sorted list to reflect its primary accessibility. The resulting ordered set of edge nodes accessible to u_h is denoted as:

$$\mathcal{C}(u_h) = \{u_h(1), \ldots, u_h(j'), \ldots, u_h(|\mathcal{C}(u_h)|)\}, \tag{7}$$

where $u_h(1)$ corresponds to the directly connected edge node, and subsequent nodes $u_h(j')$ are sorted in increasing order of the user-edge distance $d_{hj'}$.

3) User Request Model: Each user u_h generates a sequence of content requests based on real-world datasets. The request behavior of all users is captured by a binary matrix Q, defined as:

$$Q = \{q_{hi} \mid q_{hi} \in \{0, 1\}, h = 1, \ldots, H, i \in \mathcal{K}\}, \tag{8}$$

where $q_{hi} = 1$ indicates that user u_h requests content i.

4) Content Popularity: The popularity of content i at each edge node c_j is denoted by p_{ij}^t, representing the request ratio for the content within a specific time slot t. Notably, this value is normalized to lie within the range $[0, 1]$.

$$P^t = \{p_{ij}^t \mid p_{ij}^t \in [0, 1], i \in \mathcal{K}, j = 1, \ldots, J, t = 1, \ldots, T\}. \tag{9}$$

5) Content Recommendation Model: CP recommends content to the user u_h based on their preferences. The recommendation strategy is represented by the following matrix:

$$M = \{m_{hi} \mid m_{hi} \in \{0, 1\}, h = 1, \ldots, H, i \in \mathcal{K}\}, \tag{10}$$

where $m_{hi} = 1$ indicates that content i is recommended to user u_h, and $m_{hi} = 0$ otherwise. For each user u_h, the number of recommended items is constrained by N_h. This constraint is expressed as:

$$\sum_{i \in \mathcal{K}} m_{hi} \leq N_h, \quad \forall h = 1, \ldots, H. \tag{11}$$

B. Utility Analysis and Problem Formulation

1) User Utility Function: Users derive a bonus from retrieving content efficiently from edge nodes. The user bonus function is expressed as:

$$B_{u_h,i}^{\text{user}}(X,K)=\sum_{j'=1}^{|\mathcal{C}(u_h)|}\left[q_{hi}\phi(k_{hj'})x_{ij'}\prod_{l=1}^{j'-1}(1-x_{il})\right],\quad(12)$$

where $x_{ij'}\prod_{l=1}^{j'-1}(1-x_{il})$ ensures that content i is fetched from the first available edge node $c_{j'}$ in the user's access list. The function $\phi(k_{hj'})$ represents the influence of the user-to-edge node topology k_{hj} on satisfaction. It is defined as:

$$\phi(k_{hj'})=\sum_{j=1}^{J}k_{hj}\prod_{o=1}^{j-1}(1-k_{ho})\gamma_s\left(1-\hat{d}_{hj}+\frac{1}{\hat{g}_{jj'}+1}\right),\quad(13)$$

where $\sum_{j=1}^{J}k_{hj}\prod_{o=1}^{j-1}(1-k_{uo})$ identifies the edge node c_j to which user u_h is directly connected, γ_s is the satisfaction-price refraction coefficient, $\hat{d}_{hj}$ is the normalized distance between user u_h and edge node c_j, and $\hat{g}_{jj'}$ is the normalized distance between edge nodes c_j and $c_{j'}$. The normalized distance-based satisfaction incorporates two parts: the term $1-\hat{d}_{hj}$ increases as the distance between user u_h and node c_j decreases, while the term $\frac{1}{\hat{g}_{jj'}+1}$ decreases as the distance between the caching node $c_{j'}$ (storing content i) and the user's current connection node c_j increases. Together, these two parts determine the user's overall bonus.

The cost incurred by users when retrieving content is given by:

$$V_{u_h,i}^{\text{user}}=q_{hi}\delta_i\lambda_{\text{MEC}},\quad(14)$$

where λ_{MEC} represents the monetary per GB of content transmission charged by the edge nodes.

The overall user utility function combines bonus and cost:

$$U_{\text{user}}(X,K)=\sum_{u_h\in\mathcal{U}}\sum_{i\in\mathcal{K}}\left[B_{u_h,i}^{\text{user}}(X,K)-V_{u_h,i}^{\text{user}}\right].\quad(15)$$

The user's objective is to maximize their utility, which is represented as:

$$\max_{X,K} U_{\text{user}}(X,K),\quad(16a)$$

$$\text{s.t.}\quad \sum_{i\in\mathcal{K}}\delta_i x_{ij}\le s_j,\quad \forall i,j,\quad(16b)$$

$$\sum_{u_h\in\mathcal{U}}k_{hj}=1,\quad \forall h,j,\quad(16c)$$

$$x_{ij},k_{hj}\in\{0,1\},\quad \forall i,j,h.\quad(16d)$$

2) Content Popularity Prediction: At each edge node, users generate varying content preferences based on their individual demands, leading to differences in content popularity. To optimize caching strategy, it is essential to predict content popularity and improve the accuracy of global predictions. The loss function is defined as:

$$L=\sum_{i\in\mathcal{K}}\sum_{c_j\in\mathcal{C}}\left(p_{ij}^t-{p_{ij}^t}^*\right)^2,\quad(17)$$

where ${p_{ij}^t}^*$ represents the actual popularity of content i at node u_h at time slot t.

3) Edge Nodes Utility Function: Edge nodes benefit from user payments for content delivery. This benefit is defined as:

$$S_{u_h,i}^{\text{MEC}}=V_{u_h,i}^{\text{user}}.\quad(18)$$

The cost for edge nodes comprises two components based on whether the requested content is cached within edge nodes. The first component is the cost of transmitting cached content to users. When the requested content i is cached at edge node $c_{j'}$, the transmission cost includes two parts: the energy consumption for transferring content i from the caching node $c_{j'}$ to the user's current connection node c_j via a wired network, and the energy consumption for downloading the content from node c_j to the user u_h via a wireless network. This cost is expressed as:

$$V_{u_h,i}^{\text{MEC,cached}}(X,K)=\sum_{j'=1}^{|\mathcal{C}(u_h)|}\left[q_{hi}\gamma_e E_{\text{overall}}(\delta_i,K)x_{ij'}\prod_{l=1}^{j'-1}(1-x_{il})\right],\quad(19)$$

where γ_e is the energy-price refraction coefficient, and

$$E_{\text{overall}}(\delta_i,K)=\sum_{j=1}^{J}k_{hj}\prod_{o=1}^{j-1}(1-k_{ho})\left[\frac{\delta_i}{r_{j'}}P_{\text{wired}}g_{jj'}+\frac{\delta_i}{r_j}P_{\text{wireless}}(d_{hj})^{\alpha}\right],\quad(20)$$

with $r_{j'}$ representing the transmission rate of edge node $c_{j'}$ (GB/s), P_{wired} denoting the baseline wired power consumption (W/GB/km), P_{wireless} denoting the baseline wireless power consumption (W/GB/km), and α being the path loss exponent.

The second component is the cost of forwarding non-cached content, which includes the energy consumption for transferring content i from the CP to edge node c_j via a wired network and for downloading it from node c_j to user u_h via a wireless network. The cost is expressed as:

$$V_{u_h,i}^{\text{MEC,non-cached}}(X,M)=\prod_{j'=1}^{|\mathcal{C}(u_h)|}(1-x_{ij'})\left[q_{hi}\delta_i\lambda_{\text{CP}}\theta(m_{hi})+q_{hi}\gamma_e\frac{\delta_i}{r_{j'}}P_{\text{wireless}}(d_{hj'})^{\alpha}\right],\quad(21)$$

where λ_{CP} is the cost per GB charged by the content provider for content transmission. To incentivize caching of newly popular content, the CP employs a recommendation-driven discount mechanism that reduces λ_{CP} for highly recommended content. The function $\theta(m_{hi})$ is a discount function of m_{hi}, defined as:

$$\theta(m_{hi})=m_{hi}\rho+(1-m_{hi}),\quad(22)$$

where ρ is the discount factor, which applies only when recommended content is requested.

However, due to the vast and discrete solution space, directly solving the optimization target X is computationally infeasible. Content popularity serves as a practical basis for initial caching decisions, which is embedded into X as a heuristic, offering a foundational rule for decision-making. Based on this, the solution can be iteratively refined to achieve stable and efficient outcomes.

To incorporate content popularity into the caching strategy, we define a caching function as follows:

$$x_{ij} = \Phi(w_{ij} p_{ij}^t), \tag{23}$$

where $\Phi(\cdot)$ is a piecewise function dependent on the content popularity p_{ij} and weight w_{ij}. Substituting this function into the cost equations (19) and (21), we obtain the updated expressions:

$$V_{u_h,i}^{\text{MEC,cached}}(W,K) = \sum_{j'=1}^{|\mathcal{C}(u_h)|} \left[q_{hi}\gamma_e E_{\text{overall}}(\delta_i, K)\Phi(w_{ij'}p_{ij'}^t) \prod_{l=1}^{j'-1} \left(1-\Phi(w_{il}p_{il}^t)\right) \right], \tag{24}$$

$$V_{u_h,i}^{\text{MEC,non-cached}}(W,M) = \prod_{j'=1}^{|\mathcal{C}(u_h)|} \left(1-\Phi(w_{ij'}p_{ij'}^t)\right) \left[q_{hi}\delta_i\lambda_{\text{CP}}\theta(m_{hi}) + q_{hi}\gamma_e \frac{\delta_i}{r_{j'}} P_{\text{wireless}}(d_{hj})^\alpha \right]. \tag{25}$$

Here, we retain X as a core modeling element for ease of understanding the overall framework. The parameters related to the content provider, such as W, will be elaborated further in the section IV.

The total cost for edge nodes is the sum of the two components:

$$V_{u_h,i}^{\text{MEC}}(X,K,M) = V_{u_h,i}^{\text{MEC,cached}}(X,K) + V_{u_h,i}^{\text{MEC,non-cached}}(X,M). \tag{26}$$

The utility function for edge nodes is defined as:

$$U_{\text{MEC}}(X,K,M) = \sum_{u_h \in \mathcal{U}} \sum_{i\in\mathcal{K}} \left[S_{u_h,i}^{\text{MEC}} - V_{u_h,i}^{\text{MEC}}(X,K,M) \right]. \tag{27}$$

The objective of edge nodes is to maximize their utility, which is represented as:

$$\max_{X,K,M} U_{\text{MEC}}(X,K,M), \tag{28a}$$
$$\text{s.t.} \quad \sum_{i\in\mathcal{K}} \delta_i x_{ij} \le s_j, \quad \forall i,j, \tag{28b}$$
$$\sum_{u_h\in\mathcal{U}} k_{hj} = 1, \quad \forall h,j, \tag{28c}$$
$$\sum_{i\in\mathcal{K}} m_{hi} \le N_h, \quad \forall h,i, \tag{28d}$$
$$x_{ij}, k_{hj}, m_{hi} \in \{0,1\}, \quad \forall i,j,h. \tag{28e}$$

4) Content Provider Utility Function: The benefit to the CP arises when it serves content requests missed by edge nodes. The benefit function is given by:

$$S_{u_h,i}^{\text{CP}}(X,M) = [q_{hi}\delta_i\lambda_{\text{CP}}\theta(m_{hi}) + \varepsilon(m_{hi})] \prod_{j'=1}^{|\mathcal{C}(u_h)|} (1-x_{ij'}), \tag{29}$$

where $\varepsilon(m_{hi}) = \sqrt{m_{hi}}\tau_{hi}$, in which τ_{hi} denotes the estimated advertisement revenue generated through recommending content i to user u_h. The value of τ_{hi} is obtained via a collaborative filtering model that predicts user preference for unrequested content, thereby representing the expected gain from content recommendation.

The cost incurred by the CP for delivering the requested content over wired links is:

$$V_{h,i}^{\text{CP}}(X) = q_{hi}\gamma_e \frac{\delta_i}{r_{\text{CP}}} P_{\text{wired}} d_{\text{CP},\mathcal{C}} \prod_{j'=1}^{|\mathcal{C}(u_h)|} (1-x_{ij'}), \tag{30}$$

where r_{CP} denotes the transmission rate of the CP (GB/s), P_{wired} represents the baseline power consumption per unit distance in wired networks (W/GB/km), and $d_{\text{CP},\mathcal{C}}$ is the average distance between the CP and the set of edge nodes $\mathcal{C}$ (km).

We further define the bonus for the CP, which quantifies the tradeoff between content directly recommended by the CP and content delivered via collaborate edge nodes that cache the requested content i. This metric bridges the gap between direct recommendation and collaborate caching, ensuring an effective balance. The bonus is defined as:

$$B_{u_h,i}^{\text{CP}}(X,K,M) = \sum_{j=1}^{J} q_{hi}k_{hj}(1-x_{ij}) \sum_{j'=1, j'\ne j}^{|\mathcal{C}(u_h)|} x_{ij'} \left[\gamma_e \frac{\delta_i}{r_{j'}} P_{\text{wired}} g_{jj'} - (\delta_i\lambda_{\text{CP}}\theta(m_{hi}) + \varepsilon(m_{hi}) - \delta_i\gamma_e P_{\text{wired}} d_{\text{CP},\mathcal{C}}) \right]. \tag{31}$$

The utility function for the CP combines the benefit, cost, and bonus functions:

$$U_{\text{CP}}(X,K,M) = \sum_{u_h\in\mathcal{U}} \sum_{i\in\mathcal{K}} \left[S_{u_h,i}^{\text{CP}}(X,M) - V_{u_h,i}^{\text{CP}}(X) + B_{u_h,i}^{\text{CP}}(X,K,M) \right]. \tag{32}$$

The objective of CP is to maximize its utility, which is represented as:

$$\max_{X,K,M} U_{\text{CP}}(X,K,M), \tag{33a}$$
$$\text{s.t.} \quad \sum_{i\in\mathcal{K}} \delta_i x_{ij} \le s_j, \quad \forall i,j, \tag{33b}$$
$$\sum_{u_h\in\mathcal{U}} k_{hj} = 1, \quad \forall h,j, \tag{33c}$$
$$\sum_{i\in\mathcal{K}} m_{hi} \le N_h, \quad \forall h,i, \tag{33d}$$
$$x_{ij}, k_{hj}, m_{hi} \in \{0,1\}, \quad \forall i,j,h. \tag{33e}$$

IV. Game Theory Empowered Carbon-Intelligent Federated Multi-edge Caching Framework

A. Light Weight Federated Content Popularity Prediction

In real-world scenarios, content popularity often follows a Zipf's distribution [23], such as a long-tail distribution, which has been widely validated on mainstream video platforms. Under Zipf's distribution, a small fraction of content captures the majority of user attention, while a larger fraction receives relatively little engagement. Existing methods for content popularity prediction predominantly rely on Recurrent Neural Network (RNN) or their variants [24]. However, when predicting less popular content with similar levels of popularity, these methods often struggle to define clear boundaries between popular and less popular content, leading to potential confusion. To address these limitations, the Temporal Convolutional Network (TCN), based on convolutional neural networks, has emerged as an alternative to recurrent architectures. Unlike RNNs, where subsequent time steps depend on the completion of previous ones, TCN enables parallel computation across time steps, significantly improving efficiency. Additionally, it shared filters across layers reduce memory consumption, making it highly effective for long-sequence processing. By enhancing computational efficiency and minimizing resource usage, TCN contributes to lower energy consumption and reduced carbon emissions.

Specifically, the input to the TCN is the feature vector of content popularity from time $t-T$ to t, denoted as $\{Z^{t-T},\ldots,Z^{t}\}$, where $Z^t=\{p^t_{1j},\ldots,p^t_{n_j j}\}$ represents the set of content popularity features for all content on edge node c_j at time slot t, and n_j denotes the number of contents on c_j. The expected output is a vector $\{Y^{t+1},\ldots,Y^{t+T'}\}$, representing the set of content popularity features for all content over the next T' time slots. Here, $Y^{t+1}=\{p^{t+1}_{1j},\ldots,p^{t+1}_{n_j j}\}$. The goal is to construct an effective input-output mapping to predict future content popularity features based on historical request data.

The TCN adopts a 1-D fully convolutional network architecture, where each hidden layer has the same length as the input layer. This dense prediction approach allows TCN to capture information across the entire input sequence. The TCN primarily includes two types of convolutional structures: dilated causal convolution and residual blocks.

For dilated causal convolution, traditional causal convolution requires numerous layers or large filter sizes to expand the receptive field, which is necessary for capturing long-term dependencies. By applying dilated causal convolutions, the model achieves a larger receptive field. For a filter $\mathcal{F}=\{f_1,f_2,\ldots,f_k\}$, the dilated causal convolution y_t of $\mathcal{Z}=\{z_1,z_2,\ldots,z_T\}$ at z_t is defined as:

$$y_t=\sum_{k-1}^{K} f_k z_{t-(K-k)d}, \tag{34}$$

where d is the dilation factor and k is the filter size.

In the residual blocks, each block contains two dilated causal convolutions. After each convolution, weight normalization is applied to the filter, and dropout is added to the residual block for regularization.

To enhance TCN performance, the future content popularity evaluator balances short-term bursty trends with long-term memory, while accounting for varying priorities of content popularity at different time points [25]. Specifically, it considers the past l time slots of content popularity generated by the content popularity evaluator and the predicted short-term popularity. The final popularity at time $t+1$ for content i on edge node c_j is estimated as:

$$\bar{p}^{t+1}_{ij}=(1-\eta)p^{t+1}_{ij}+\sum_{t'=t-l+1}^{t}\eta^{t-t'+1}p^{t'}_{ij}, \tag{35}$$

where $p^{t'}_{ij}$ represents the popularity of content i at time slot t', $\bar{p}^{t+1}_{ij}$ is the predicted popularity of i at $t+1$, η is a constant $(0<\eta<1)$ that adjusts the balance between historical and recent data, and l is the length of the historical data considered. This method effectively mitigates the impact of high user dynamics, balancing short-term and long-term memory for accurate content popularity prediction.

Moreover, compared to centralized content popularity prediction methods reliant on cloud data centers, distributed approaches have gained increasing attention. FL, as an efficient edge collaboration algorithm, enables the effective aggregation of local learning models while mitigating the issue of data silos. However, with the emergence of large-scale deep learning models, the communication cost of FL has surged significantly. Therefore, exploring lightweight model transmission schemes is essential to align with the trend of low-carbon intelligent computing.

To address this challenge, we propose a novel lightweight TCN-based federated content popularity prediction framework. This framework aggregates the local TCN-based prediction models of edge nodes to generate a global prediction model, which is then distributed back to each edge node.

Specifically, the TCN-based content prediction model of each edge node is denoted as ω_j. We employ the 3LC data compression scheme to reduce the size of the uploaded local TCN-based prediction models. 3LC integrates three consecutive techniques to compress the parameter matrix of the model: 3-value quantization with sparsity multiplication, quartic encoding, and zero-run encoding, which collectively balance traffic reduction, accuracy, and computational efficiency [26]. The compression level is controlled by the sparsity parameter ν. First, ω_j is sparsified and quantized into values $\{-1,0,1\}$ to produce $\hat{\omega}_j$:

$$\hat{\omega}_j=\text{round}\left(\frac{\omega_j}{s_j}\right), \tag{36}$$

where

$$s_j=\nu\max|\omega_j|. \tag{37}$$

Next, $\hat{\omega}_j$ is compressed using Quartic Encoding (QE). Since $\hat{\omega}_j$ contains only three distinct values, it can be encoded more efficiently than a conventional 2-bit representation. In QE, $\hat{\omega}_j$ is first flattened and then zero-padded to ensure its length is a multiple of five. The sequence is then split into five equal parts and encoded into an 8-bit unsigned integer array, denoted as $\mathbf{r}$. The final step of 3LC is Zero Run Encoding (ZRE), which further compresses the data by encoding consecutive runs of common bytes using a specialized run-length encoding technique tailored for quartic encoded data. After applying ZRE, the compressed model parameters are represented as $\mathbf{r}_{\text{ZR}}$ and s_j, which are transmitted to the server. The server then performs the inverse operations to reconstruct $\hat{\omega}_j$.

Finally, Federated Learning generates a globally shared model ω^* based on the local request data volume. The aggregation process is formulated as:

$$\omega^* = \frac{1}{|\mathcal{C}|} \sum_{c_j \in \mathcal{C}} \left(\frac{\sum_{u_h \in \mathcal{U}, i \in \mathcal{K}} k_{hj} q_{hi}}{\sum_{u_h \in \mathcal{U}, i \in \mathcal{K}} q_{hi}} \hat{\omega}_j \right), \tag{38}$$

where $\mathcal{C}$ represents the set of participating edge nodes, $\mathcal{U}$ denotes the set of users, and $\mathcal{K}$ represents the set of content items. The parameters q_{hi} indicates the local request frequency, ensuring that the global model is optimized based on the observed content popularity patterns at different edge nodes.

B. Two-Stage Game Theoretic Design

From the utility objectives of users (16), edge nodes (28), and the CP (33), it is evident that their respective objective functions are intrinsically intertwined, as they all rely on the variables X, K, and M. This interdependence establishes a complex coupling relationship between them. To address these intricate non-linear optimization problems, we thus propose a two-stage game theoretic algorithm. It systematically disentangles the coupled decision-making processes and ensures an optimal balance among stakeholders.

We first analysis the interactions between users and edge nodes, abstracting their interplay as the first stage game. In the absence of the CP's influence, it becomes evident that these two decision-making processes manifest as coupled 0-1 nonlinear optimization problems. Within this intricate coupling, the variable X is dictated by the edge nodes, whereas the variable K is determined by the users. Therefore, this interdependent structure enables us to formally characterize the game by:

$$U_{\text{user}}(X^*, K^*) \geq U_{\text{user}}(X^*, K), \tag{39}$$

$$U_{\text{MEC}}(X^*, K^*) \geq U_{\text{MEC}}(X, K^*), \tag{40}$$

where K^* represents the optimal user-to-edge topology and X^* is the optimal caching strategy.

It is important to emphasis that X^* is not solely pertinent to the user and edge node, but also intricately linked to the CP. Based on this dual dependency, we devise a second stage that addresses the interactions between these key stakeholders. One of the second stage aims is to derive a potentially optimal X to guide the first stage to obtain the K^*.

In the second stage, we then shift to the intricate coupling between edge nodes and the CP. We assume that K^*, *i.e.,* topology is a known parameter by the first stage. Within this coupled problem, the edge nodes and the CP engage in an interaction, jointly optimizing the decision variables X and M, *i.e.,* recommendation strategy and caching strategy. Similarly, by analysing the caching strategy X and recommendation strategy M in (28) and (33), it can be observed that this problem is a coupled discrete nonlinear optimization problem. The objective of the second game is formulated as:

$$U_{\text{MEC}}(X^*, K^*, M^*) \geq U_{\text{MEC}}(X, K^*, M^*), \tag{41}$$

$$U_{\text{CP}}(X^*, K^*, M^*) \geq U_{\text{CP}}(X^*, K^*, M). \tag{42}$$

As X undergoes adjustments, the interdependence between decision variables necessitates a return to the first stage, prompting a reevaluation of the interactions. This iterative process continues as the game progresses, dynamically refining the equilibrium until the three-party system ultimately converges to a stable solution. In the following, we provide a comprehensive exposition of the two-stage game algorithm, detailing its procedural flow and convergence mechanism.

Section IV-C and section IV-D will provide in-depth analyse of the two stages of the game theoretic design whereas section IV-E describes the interactions between the two stages of the game.

C. Stage 1: Matching Game Theoretic approach for Topolgy

1) Game Design: Given that a single edge node serves multiple users and there are multiple edge nodes in the network, we formulate the first-stage interaction as a matching game. It aims to capture the user-to-edge node topology relationship between users and edge nodes in the first stage in terms of known X. The X can be obtained in the second stage and in turn re-guide this stage. The foundation of this approach lies in the stable matching theory, originally introduced by Gale and Shapley through the stable marriage problem [27]. They proposed the deferred acceptance algorithm to achieve a stable matching between two sets of entities (e.g., men and women) by iterating through a process of proposal and rejection. However, the classical deferred acceptance algorithm is specifically designed for solving one-to-one matching problems. In our context of optimizing the user-to-edge node topology, a one-to-many matching model is required due to the capacity constraints of edge nodes, which can serve multiple users simultaneously.

To address this, we extend the deferred acceptance algorithm to accommodate one-to-many matching scenarios by incorporating two key factors: the maximum workload capacity of each edge node and the connection constraints for each user. The rules and definitions of stable matching are redefined to ensure that the resulting topology remains efficient and stable. This extension allows for an effective mapping of users to edge nodes, balancing users' utility and edge nodes' utility while maintaining stability in the matching process.

To support the matching process, we first construct the utility lists for both sides. Specifically, we evaluate the user and edge node utility functions under single-point topology changes while keeping the caching and recommendation strategies fixed. Based on these utility lists, we design three rules for the stable user-to-edge node matching algorithm. These rules extend the traditional deferred acceptance algorithm to handle the one-to-many matching scenario and are defined as follows:

1) Invitation Rule: Each user sends connection requests to edge nodes in order of their utility list until they receive a response or exhaust their list.
2) Acceptance Rule: An edge node tentatively accepts connection requests from users based on its utility list, up to its maximum connection capacity. If the edge node

receives more requests than it can handle, it temporarily accepts the users with higher utility.

3) Rejection Rule: In case an edge node reaches its connection capacity, it rejects lower-ranked users in favor of higher-ranked users who submit new requests. Rejected users then proceed to send requests to the next edge node on their utility list.

These rules ensure that the user-to-edge node matching process respects the capacity constraints of edge nodes while maintaining stability in the final matching outcome. The specific algorithm is shown in Algorithm 1.

Algorithm 1: Stable User-to-Edge Node Matching via Extended Deferred Acceptance

---

Input: Set of users $\mathcal{U}$, set of edge nodes $\mathcal{C}$, utility lists P_{u_h} for each $u_h \in \mathcal{U}$, utility lists P_{c_j} for each $c_j \in \mathcal{C}$, maximum connection capacity C_j for each $c_j \in \mathcal{C}$.

Output: Stable user-to-edge node matching matrix K

Initialization: Initialize $K = \{k_{hj} = 0 \mid h = 1, \ldots, H, j = 1, \ldots, J\}$.
Initialize proposal index $\nu_h = 1$ for each user u_h to track the next edge node to propose to in P_{u_h}.
Initialize proposal sets $\mathcal{Q}_j = \emptyset$ for each edge node c_j.

while *there exists at least one unmatched user $u_h \in \mathcal{U}$ and $\nu_h \le |P_{u_h}|$* **do**

    Phase 1: Invitation;
    foreach *unmatched user $u_h \in \mathcal{U}$* ***in parallel*** **do**
        Let $c_j = P_{u_h}[\nu_h]$, the edge node with the highest utility in P_{u_h} not yet proposed to;
        Add u_h to c_j's proposal set: $\mathcal{Q}_j \leftarrow \mathcal{Q}_j \cup \{u_h\}$;
        Increment ν_h: $\nu_h \leftarrow \nu_h + 1$;
    end
    Phase 2: Accept and Reject;
    foreach *edge node $c_j \in \mathcal{C}$* ***in parallel*** **do**
        Form candidate set: $\mathcal{R}_j = \mathcal{Q}_j \cup \{u_h \mid k_{hj} = 1\}$;
        Select top C_j users: $\mathcal{S}_j =$ Top-C_j users in $\mathcal{R}_j$ per P_{c_j};
        Update matching matrix: $k_{hj} = 1\, \forall u_h \in \mathcal{S}_j$;
        Reset proposals: $\mathcal{Q}_j \leftarrow \emptyset$;
    end
    Phase 3: Invitation;
    foreach *user u_h rejected by c_j in this round* ***in parallel*** **do**
        Continue to the next edge node in P_{u_h} with ν_h;
    end

end

return K;

---

2) Toy Example of One-to-Many Stable Matching: To illustrate the proposed stable user-to-edge node matching algorithm, consider a system with five users $(u_1, u_2, u_3, u_4, u_5)$ and two edge nodes (c_1, c_2). Each user has a utility list P_{u_h} for the edge nodes, and each edge node has a utility list P_{c_j} for the users. The utility are as follows:

- Users' Utility Lists:
  - u_1 prefers c_1 over c_2 ($P_{u_1} = [c_1, c_2]$).
  - u_2 prefers c_1 over c_2 ($P_{u_2} = [c_1, c_2]$).
  - u_3 prefers c_2 over c_1 ($P_{u_3} = [c_2, c_1]$).
  - u_4 prefers c_1 over c_2 ($P_{u_4} = [c_1, c_2]$).
  - u_5 prefers c_2 over c_1 ($P_{u_5} = [c_2, c_1]$).
- Servers' Utility Lists:
  - c_1 ranks users in the order u_5, u_2, u_3, u_4, u_1 ($P_{c_1} = [u_5, u_2, u_3, u_4, u_1]$).
  - c_2 ranks users in the order u_2, u_4, u_1, u_3, u_5 ($P_{c_2} = [u_2, u_4, u_1, u_3, u_5]$).

Assume that c_1 has a maximum workload capacity of 2 and c_2 has a capacity of 3. The goal is to determine a stable matching where users are assigned to edge nodes, respecting the capacity constraints and the preferences of both users and edge nodes.

The matching process begins with all users being unmatched. In the first round, each user selects the edge node with the highest utility from their utility list. Specifically, u_1, u_2, and u_4 propose to c_1, while u_3 and u_5 propose to c_2. The edge nodes then review the proposals and tentatively accept the users based on their utility lists, up to their capacity limits. Edge node c_1 receives proposals from u_1, u_2, and u_4. Since c_1 can accept at most two users, it selects u_2 and u_4, and rejects u_1. Edge node c_2 receives proposals from u_3 and u_5. Since c_2 has a capacity of 3, it tentatively accepts both u_3 and u_5.

In the second round, u_1, who was rejected by c_1, now proposes to c_2, their next preferred edge node. c_2 now has three proposals: u_3, u_5, and u_1. Since c_2 has a capacity of 3, it tentatively accepts all three users.

At this point, all users are tentatively matched, and the algorithm terminates. The final stable matching is as follows: c_1 is matched with u_2 and u_4, while c_2 is matched with u_3, u_5, and u_1.

D. Stage 2: Stackelberg Game Theoretic Approach for Recommendation Strategy and Caching Strategy

In the second stage, given the K^*, we analyse the optimal selection of X and M, which subsequently influences and refines the K^* decision-making process in the first stage. Recognizing that each unique content is exclusively provided by a single CP distributed across multiple edge nodes, in this stage, we formulate the optimization of recommendation strategy and caching strategy as a Stackelberg (Leader-Follower) game between edge nodes and CPs. Furthermore, to achieve the Nash Equilibrium (NE) and enhance the computational efficiency of this game, we relax the binary decision variables of X and M into continuous variables $\tilde{X}$ and $\tilde{M}$ and project after the second stage game. The objective is to establish a hierarchical decision-making process that maximizes the utilities of both entities while ensuring recommendation efficiency and optimal content placement.

The Stackelberg framework can be described as follows:

1) Leader's Problem (Edge Nodes): The edge nodes determine the optimal caching strategy $\tilde{X}^*$ by solving the following optimization problem:

$$\tilde{X}^* = \arg\max_{\tilde{X}} U_{\text{MEC}}(\tilde{X}, K^*, \tilde{M}^*(\tilde{X})), \tag{43}$$

where $U_{\text{MEC}}(\tilde{X}, K^*, \tilde{M}^*(\tilde{X}))$ represents the utility of the edge nodes, which depends on the caching strategy $\tilde{X}$, the refined user-to-edge node topology strategy K^*, and the content providers' best-response recommendation strategy $\tilde{M}^*(\tilde{X})$.

2) Follower's Problem (Content Providers): Given a caching strategy $\tilde{X}^*$ by the edge nodes, the content providers determine their optimal recommendation strategy $\tilde{M}^*$ by solving:

$$\tilde{M}^* = \arg\max_{\tilde{M}} U_{\text{CP}}(\tilde{X}^*, K^*, \tilde{M}), \tag{44}$$

where $U_{\text{CP}}(\tilde{X}^*, K^*, \tilde{M})$ represents the utility of the content providers, which depends on the initial or refined caching strategy $\tilde{X}^*$, refined user-to-edge node topology strategy K^*, and the recommendation strategy $\tilde{M}$.

3) Joint Optimization Objective: The Stackelberg game aims to achieve joint optimization of caching and recommendation strategy, which expressed as (41)-(42).

To solve the Stackelberg game, we first address the content providers' problem by deriving the best-response recommendation strategy $\tilde{M}^*$ for a given caching strategy $\tilde{X}^*$ and user-to-edge node topology strategy K^*. This involves solving the follower's optimization problem to maximize the content providers' utility while satisfying the recommendation constraints. Once the optimal recommendation strategy $\tilde{M}^*$ is obtained, we substitute it into the edge nodes' utility function. The edge nodes then solve their optimization problem to determine the optimal caching strategy $\tilde{X}^*$, considering the anticipated response from the content providers.

The expression for optimization problem (44) is given as:

$$\tilde{M}^* = \arg\min_{\tilde{M}} \sum_{u_h \in \mathcal{U}} \sum_{i \in \mathcal{K}} \Big[- S^{\text{CP}}_{u_h,i}(\tilde{X}, \tilde{M}) + V^{\text{CP}}_{u_h,i}(\tilde{X}) \tag{45a}$$

$$- B^{\text{CP}}_{u_h,i}(\tilde{X}, K, \tilde{M}) \Big]. \tag{45b}$$

where the binary variables M are relaxed into continuous variables, i.e., $\tilde{m}_{hi} \in [0, 1]$.

Next, define the objective function as:

$$F(\tilde{m}_{hi}) = \Big[- S^{\text{CP}}_{u_h,i}(\tilde{X}, \tilde{M}) + V^{\text{CP}}_{u_h,i}(\tilde{X}) - B^{\text{CP}}_{u_h,i}(\tilde{X}, K, \tilde{M}) \Big], \tag{46}$$

Theorem 1: The best response function of $\tilde{m}_{hi}$ is:

$$\tilde{m}^*_{hi} = F(K^*, \tilde{X}). \tag{47}$$

Proof: Taking the derivative of (45) with respect to $\tilde{m}_{hi}$, we obtain:

$$\frac{\partial F(\tilde{m}_{hi})}{\partial \tilde{m}_{hi}} = q_{hi} \left[\delta_i \lambda_{\text{CP}} (1-\rho) - \frac{1}{2} \tau_{hi} \tilde{m}_{hi}^{-\frac{1}{2}} \right] \prod_{j'=1}^{|\mathcal{C}(u_h)|} (1 - \tilde{x}_{ij'})$$
$$+ \sum_{j=1}^{J} \sum_{j'=1, j' \neq j}^{|\mathcal{C}(u_h)|} q_{hi} k_{hj} (1 - \tilde{x}_{ij}) \tilde{x}_{ij'} \left[\delta_i \lambda_{\text{CP}} (1-\rho) - \frac{1}{2} \tau_{hi} \tilde{m}_{hi}^{-\frac{1}{2}} \right]. \tag{48}$$

The second derivative of $F(\tilde{m}_{hi})$ with respect to $\tilde{m}_{hi}$ is:

$$\frac{\partial^2 F(\tilde{m}_{hi})}{\partial \tilde{m}_{hi}^2} = \frac{1}{4} q_{hi} \delta_i \tau_{hi} \tilde{m}_{hi}^{-\frac{3}{2}} \prod_{j'=1}^{|\mathcal{C}(u_h)|} (1 - \tilde{x}_{ij'})$$
$$+ \sum_{j=1}^{J} \sum_{j'=1, j' \neq j}^{|\mathcal{C}(u_h)|} q_{hi} k_{hj} (1 - \tilde{x}_{ij}) \tilde{x}_{ij'} \frac{1}{4} \tau_{hi} \tilde{m}_{hi}^{-\frac{3}{2}}. \tag{49}$$

Since the second derivative satisfies $\frac{\partial^2 F(\tilde{m}_{hi})}{\partial \tilde{m}_{hi}^2} \geq 0$, $F(\tilde{m}_{hi})$ is convex. Solving $\frac{\partial F(\tilde{m}_{hi})}{\partial \tilde{m}_{hi}} = 0$ is not straightforward. However, for given values of $q_{hi}, \delta_i, \tau_{hi}$, and ρ, the solution depends only on the caching strategy X and optimal user-to-edge node topology K^* from section IV-C. Therefore, we obtain:

$$\tilde{m}^*_{hi} = F(K^*, X) \tag{50}$$

By the Debreu-Glicksberg-Fan theorem, a pure strategy NE exists as the strategy space is compact and $F(\tilde{m}_{hi})$ is concave in $\tilde{m}_{hi}$.

After obtain the optimal recommendation strategy, now we could find the caching strategy. The reformulated optimization problem for the caching strategy $\tilde{X}^*$ is expressed as follows:

$$\tilde{X}^* = \arg\min_{\tilde{X}} \sum_{u_h \in \mathcal{U}} \sum_{i \in \mathcal{K}} q_{hi} \delta_i \Bigg[- \lambda_{\text{MEC}}$$
$$+ \sum_{j'=1}^{|\mathcal{C}(u_h)|} \gamma_e P_{\text{overall}}(k_{hj'}) \tilde{x}_{ij'} \prod_{l=1}^{j'-1} (1 - \tilde{x}_{il})$$
$$+ \prod_{j'=1}^{|\mathcal{C}(u_h)|} (1 - \tilde{x}_{ij'}) \Big(\lambda_{\text{CP}} \theta(F(K^*, \tilde{X})) + \frac{1}{r'_j} \gamma_e P_{\text{wireless}}(d_{hj'})^{\alpha} \Big) \Bigg]. \tag{51}$$

Here, the binary variables x_{ij} have been relaxed into continuous variables $\tilde{x}_{ij} \in [0, 1]$.

Furthermore, according to (23), x_{ij} needs to be converted to incorporate content popularity. Since x has been relaxed into continuous variables, the embedding function Φ is transformed into a sigmoid-like function $\tilde{\Phi}$ to facilitate gradient-based optimization, defined as:

$$\tilde{\Phi}(p_{ij}) = \frac{1}{1 + e^{-a(w_{ij} p_{ij} - b)}}, \tag{52}$$

where $a > 0$ controls the steepness of the function and b is a bias term to center the output around 0.5. This smooth approximation preserves differentiability and guides the optimizer toward values near 0 or 1. This embedding ensures that the relaxed variables $x_{ij} \in [0, 1]$ can be effectively optimized using stochastic gradient descent (SGD).

Then, the reconstructed optimization problem is expressed as:

$$W^* = \arg\min_{W} \sum_{u_h \in \mathcal{U}} \sum_{i \in \mathcal{K}} q_{hi} \delta_i \Bigg[- \lambda_{\text{MEC}}$$
$$+ \sum_{j'=1}^{|\mathcal{C}(u_h)|} \gamma_e P_{\text{overall}}(k_{hj'}) \tilde{\Phi}(w_{ij'} p^t_{ij'}) \prod_{l=1}^{j'-1} (1 - \tilde{\Phi}(w_{il} p^t_{il})) +$$
$$\prod_{j'=1}^{|\mathcal{C}(u_h)|} (1 - \tilde{\Phi}(w_{ij'} p^t_{ij'})) \Big(F(K^*, W, P^t) + \frac{1}{r'_j} \gamma_e P_{\text{wireless}}(d_{hj'})^{\alpha} \Big) \Bigg]. \tag{53}$$

In the Stackelberg game framework, collaboration between the leader and the follower plays a pivotal role in iteratively refining the solution towards optimality. Each iteration involves solving a optimization problem, progressively improving the results. To ensure convergence with minimal error while maintaining feasibility, it is essential to define a robust stopping criterion. This criterion can be formulated based on the Frobenius norm, which measures the overall difference between matrices in successive iterations. Specifically, the stopping criterion is expressed as:

$$\|\tilde{M}^{(iter.+1)} - \tilde{M}^{(iter.)}\|_F \le \epsilon_{\tilde{M}}, \tag{54}$$

$$\|\tilde{X}^{(iter.+1)} - \tilde{X}^{(iter.)}\|_F \le \epsilon_{\tilde{X}}, \tag{55}$$

where $\|\cdot\|F$ denotes the Frobenius norm, calculated as the square root of the sum of the squared differences between corresponding matrix elements. The thresholds $\epsilon_{\tilde{M}}$ and $\epsilon_{\tilde{X}}$ are predefined to control the convergence of the recommendation matrix $\tilde{M}$ and the caching strategy matrix $\tilde{X}$, respectively.

In the process of solving the optimization problem, for computational convenience, the variables $\tilde{X}$ and $\tilde{M}$ are relaxed to continuous variables within the interval $[0, 1]$. However, in practical cache and recommendation strategy, the decision variables must be binary, that is, $x_{ij} \in \{0, 1\}$ and $m_{hi} \in \{0, 1\}$. Consequently, a randomized rounding approach is employed to convert the continuous solutions $\tilde{X}^*$ and $\tilde{M}^*$ into discrete binary solutions.

More specifically, the randomized rounding technique uses the fractional part of the continuous solution, denoted as $\tilde{x}_{ij}^f$, as the rounding probability. With probability $\tilde{x}_{ij}^f$, the value $\tilde{x}_{ij}$ is rounded up to $\lfloor \tilde{x}_{ij} \rfloor + 1$, and with probability $1 - \tilde{x}_{ij}^f$, it is rounded down to $\lfloor \tilde{x}_{ij} \rfloor$. Similarly, this approach is applied to the binary recovery of the recommendation strategy $\tilde{M}$.

It is important to note that the rounding process must be constrained by the problem's conditions. Specifically, the constraints on the problem are (33b)-(33e).

The rounding process, which ensures adherence to these constraints, is outlined in Algorithm 2.

E. Interaction between Two Stages of the Game Theoretic Design

Once the caching strategy X^* and recommendation strategy M^* are refined, the updated content availability and the newly requested content at edge nodes may lead users and edge nodes to reconsider their preferences, triggering an adjustment in the user-to-edge node topology K^*. This feedback loop initiates an alternating optimization mechanism, where the two stages iteratively refine their strategy until an equilibrium is achieved, at which point no participant has an incentive to unilaterally change their strategy. This iterative interaction ensures a balanced optimization of user satisfaction, caching efficiency, and overall system stability, enabling the realization of a carbon-intelligent multi-edge caching network.

V. Performance Evaluation

In this section, we introduce the experiment setup and evaluate the proposed GT-FMC through extensive comparative experiments.

Algorithm 2: Stackelberg Game for Recommendation and Caching Strategy

Input: Set of users $\mathcal{U}$, set of edge nodes $\mathcal{C}$.
Output: Optimized recommendation strategy M, caching strategy X.

```
Initialization: Compute content popularity matrix P^t
  using the TCN model. Establish initial user-to-edge
  node topology K by stable matching.
while M̃ and X̃ do not satisfy (54) and (55) do
    Phase 1: Game for Recommendation Strategy;
    Solve the convex problem (45) to obtain optimized
      recommendation strategy M̃*;
    Phase 2: Game for caching strategy;
    Convert problem (43) and reformulate as (53)
      using a sigmoid-like embedding function Φ̃;
    Apply SGD to solve (53), yielding optimized
      weights W*;
    Update caching strategy as X̃* = Φ̃(W* P^t);
end
Phase 3: Randomized Rounding;
while M̃ and X̃ do not satisfy the discrete constraints
  (33b)-(33e) do
    foreach ṽ ∈ {M̃, X̃} in parallel do
        v = { ⌊ṽ⌋ + 1,  with probability ṽ^f,
            { ⌊ṽ⌋,      with probability 1 − ṽ^f.
    end
end
return M, X;
```

A. Experimental Setup

Real-world Testbed. We set up a real-world testbed consisting of a workstation and three Jetson TX2 devices. The workstation serves as the content provider and is equipped with two NVIDIA GeForce GTX 3090 GPUs, an Intel Xeon Silver 4208 CPU (2.10GHz), and 32GB of RAM. The Jetson TX2s function as edge nodes, each featuring an NVIDIA Pascal GPU with 256 CUDA cores, and a CPU consisting of a 2-core Denver2 and a 4-core ARM Cortex-A57. All devices are connected via the same network. The communication between the workstation and Jetson TX2s is managed through a backend system built on the Message Passing Interface (MPI) framework. Additionally, the devices in the testbed run Ubuntu 18.04 OS, with CUDA v10.0 and cuDNN v7.5.0 for optimized performance.

Datasets. We use the MovieLens 1M dataset, a real-world movie rating dataset collected by GroupLens Research [28]. It contains approximately 1 million ratings provided by 6,040 users for 3,883 movies. The dataset includes labels such as anonymous userID, movieID, movie ratings, and timestamps, along with contextual information related to the users. The datasets are split into the training (70%), validation (10%), and testing (20%) sets, respectively. Additionally, the dataset is segmented into time slots on a daily basis, and a total of

250 time slots are selected for evaluation in this study.

Experiment Parameters. The parameters used in our experiment are listed in Table II.

TABLE II
EXPERIMENT PARAMETERS

Parameter	Symbol	Value
Energy-price refraction coefficient	γ_e	0.0002
Satisfaction-price refraction coefficient	γ_s	0.01
Content provider transmission rate [29]	r_{CP}	20 GB/s
Edge node transmission rate [29]	r_j	[8, 10] GB/s
Wireless transmission power consumption [29]	P_{wireless}	5 W/GB/km
Wired transmission power consumption [29]	P_{wired}	0.8 W/GB/km
Path loss exponent [30]	α	2
Content size range	δ_i	[1, 1.5] GB
Content provider transmission cost [31]	λ_{CP}	0.002 $
Edge node transmission cost [31]	λ_{MEC}	0.002 $
Recommendation discount factor in CP revenue	ρ	40%
Historical time window length [25]	l	5
Long-short term balance factor [25]	η	0.8
Error tolerance in continuous relaxation constraints	$\epsilon_{\tilde{M}}, \epsilon_{\tilde{X}}$	0.01

Comparison Approaches. We compare the TCN-based federated content popularity prediction with the optimum and five baseline methods.

- *Oracle* [32] assumes perfect knowledge of future user requests, enabling it to achieve the optimal cache hit rate given the limited cache space.
- *Random* [33] randomly selects user-requested content for proactive caching without considering content popularity or request patterns.
- *First-In-First-Out (FIFO)* [34] follows a simple eviction policy where the oldest cached content is replaced first when new content needs to be stored, regardless of its popularity or future requests.
- *Least Recently Used (LRU)* [35] prioritizes evicting the least recently requested content, maintaining frequently accessed content in the cache to improve hit rates.
- *Edge Cooperative Caching (ECC)* [36] integrates a neural collaborative filter for content popularity prediction and a greedy algorithm for content delivery.
- *TCN Prediction w/o Long-Term Memory* employs a TCN model to predict future content popularity but does not incorporate mechanisms to balance short-term fluctuations and long-term trends.

B. Experiment Results and Analysis

Federated Content Popularity Prediction. We conduct a comparative experiment on the MovieLens 1M dataset to evaluate cache hit rate variations over time slots, with each edge node allocated a cache space of 400GB. As shown in Fig. 2, the cache hit rate fluctuates over time, while the differences among the evaluated methods remain distinct. *Oracle* serves as the optimal theoretical benchmark, achieving an average cache hit rate of 60.6% by leveraging complete foresight of future content requests. In contrast, *Random* exhibits the poorest performance, achieving only 10.2% due to its uninformed caching decisions. Traditional caching strategy such as *LRU* and *FIFO*, which can efficiently handle bursty and sparse content requests, struggle to adapt to dynamic content popularity shifts. Both methods yield an average cache hit rate of 19.4%, accounting for 32% of *Oracle*'s performance.

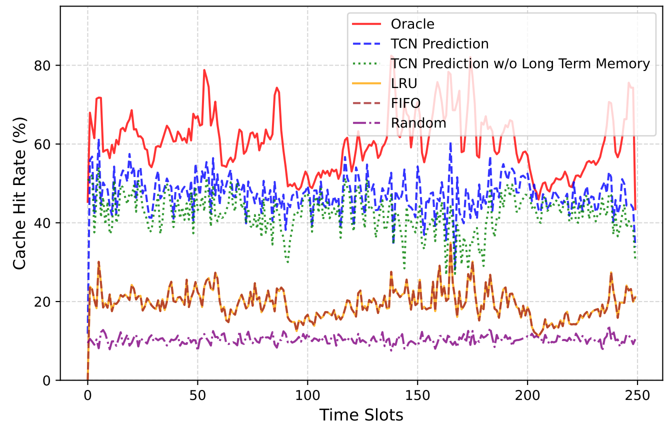

Fig. 2. Comparison between the TCN-based federated content popularity prediction and baseline methods.

In comparison, the *TCN Prediction* approach significantly enhances the cache hit rate, achieving an average of 47.2%, which corresponds to 77.9% of *Oracle*'s performance. This improvement is primarily attributed to TCN's capability to effectively capture temporal dependencies and leverage aggregated knowledge across edge nodes, enabling a more accurate prediction of user request patterns. Furthermore, the inclusion of long-term memory mechanisms plays a crucial role in enhancing prediction accuracy and stability. The variant of *TCN Prediction w/o Long-Term Memory* achieves a cache hit rate of 42.6%, which is 4.6 percentage points lower than the full *TCN Prediction* model. This performance gap highlights the importance of balancing short-term fluctuations with long-term trends in content popularity prediction, as short-term memory alone may lead to suboptimal caching decisions in highly dynamic environments.

TABLE III
COMPRESSION PERFORMANCE AND CACHE HIT RATE UNDER DIFFERENT SPARSITY LEVELS

Sparsity	Model Size (KB)		Transmission Efficiency	Cache Hit Rate
	No Comp.	With Comp.	(Compression Ratio)	(% of Oracle)
1.8	257.7	1.9	135.8×	74.3%
1.6	257.7	2.2	117.6×	75.1%
1.4	257.7	2.3	109.9×	75.9%
1.2	257.7	2.5	101.8×	76.9%
1.0	257.7	3.6	97.8×	77.9%

Compression Performance and Cache Hit Rate Analysis. Table III presents the compression performance and corresponding cache hit rates achieved under different sparsity levels using our proposed lightweight TCN-based federated content popularity prediction model. The uncompressed model size remains constant at 257.7 KB. As the sparsity parameter increases from 1.0 to 1.8, the compressed model size is significantly reduced, reaching as low as 1.9 KB at sparsity level 1.8, achieving a compression ratio of 135.8×. To better highlight the system-level benefits, we define this compression ratio as an indicator of transmission efficiency, as it directly reflects the reduction in communication overhead and energy consumption during federated model updates. Despite the aggressive compression, the cache hit rate remains relatively stable and degrades only slightly. Specifically, at sparsity 1.0, the cache hit rate reaches 77.9% of Oracle. As the sparsity increases to 1.8, the cache hit rate drops modestly to 74.3%. This demonstrates the robustness and effectiveness of our model compression strategy, which maintains high prediction accuracy even at extremely low transmission costs.

TABLE IV
TRANSMISSION ENERGY CONSUMPTION (J) COMPARISON IN GT-FMC: TCN-BASED FEDERATED CONTENT POPULARITY PREDICTION VS. BASELINE METHODS

Cache Size (GB)	Oracle	TCN Prediction	ECC	LRU	Random
400	2746.94	<u>4442.28</u>	4461.45	5484.01	6125.98
300	3530.94	<u>4761.82</u>	5215.07	5787.89	6400.38
200	4314.94	<u>5122.46</u>	5729.08	6082.27	6502.30
100	5098.94	<u>5757.50</u>	6233.78	6509.59	6572.86

Transmission Energy Consumption. The transmission energy consumption is evaluated using time slot 145 of the MovieLens 1M dataset to guide content caching strategy in GT-FMC based on different content popularity prediction methods. As the cache size decreases, the transmission energy consumption increases for all methods. The *Oracle* method serves as the theoretical benchmark, representing the optimal performance. The *Random* method performs the worst due to its lack of intelligent caching decisions, followed closely by *LRU*, which demonstrates some adaptation to content popularity variations. It is noteworthy that, despite *ECC*'s integration of collaborative filtering for cache decision guidance, it struggles in capturing rapidly changing content popularity in the time-series network. *ECC* fails to effectively model the long-term and short-term user content preferences, making it less effective than TCN-based popularity prediction. In contrast, TCN-based content popularity prediction leverages its rich knowledge of content popularity to provide efficient and accurate guidance for optimizing the caching mechanism in GT-FMC.

TABLE V
COMPARISON OF MATCHING STRATEGIES IN USER-TO-EDGE NODE ASSIGNMENT (UTILITY IN $)

Metric	Stable Matching	K-Means	Random
User Utility	1.0471	0.9571	0.6739
Edge Node Utility	1.9228	1.9227	1.8740
Total Utility	**2.9699**	**2.8798**	**2.5479**

Matching Game for Topology. We evaluate the effectiveness of the proposed matching game for user-to-edge node topology by analysing the spatial distribution of 25 users and 3 edge nodes within time slot 145 of the MovieLens 1M dataset, as shown in Fig. 3. Users are spatially distributed within the service region according to a PPP model, while the system's heterogeneous edge nodes provide caching services. It is important to note that the current caching strategy is initialized using Federated Content Popularity Prediction, while the recommendation strategy is initialized using a collaborative

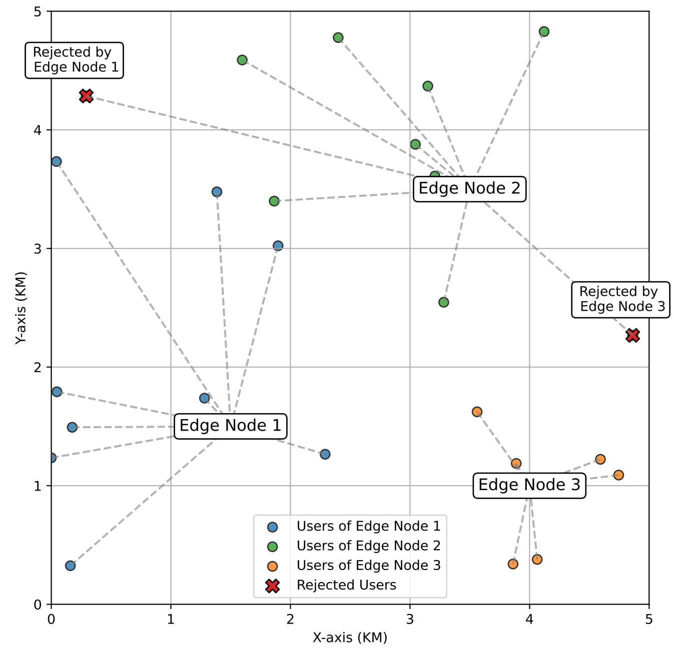

Fig. 3. Stable matching map for user-to-edge node topology.

filtering approach. We further conduct a quantitative comparison of our proposed matching strategy against two commonly used baselines: random assignment and K-means-based user clustering. As presented in Table V, our Extended Deferred Acceptance mechanism consistently achieves higher utility for both users and edge nodes. Specifically, the total system utility reaches \$2.9699 with our matching game, compared to \$2.8798 for K-means and \$2.5479 for random matching. These results verify that our method enables more effective user-to-edge associations by accounting for both user and edge preferences through Invitation, Acceptance, and Rejection rules. This approach lays the foundation for further optimization of both caching and recommendation strategy.

Stackelberg Game. We demonstrate the effectiveness of the Stackelberg game formulation through the variations in the utility of the CP, as influenced by the caching and recommendation strategy. The three-dimensional axes in the plot represent, in time slot 145 of the MovieLens 1M dataset, the caching strategy, the recommendation strategy, and the utility of the CP for content item 651, with the first two strategy being transformed into continuous variables. As discussed in the paper, the relationship between the utility of the CP and caching strategy M is convex, and for a fixed M, the relationship between the recommendation strategy X and the Utility of the CP is also convex. Notably, the utility of the CP increases with M, as the CP profits from the caching strategy, while it decreases with X, since local or collaborative node caching reduces the CP's profit. Clearly, under fixed conditions, recommending content that is not cached locally or collaboratively benefits the CP.

Collaboration between CP and Edge Nodes. We investigate the role of the critical hyperparameter ρ in the collaboration between CP and edge nodes in time slot 145 of the MovieLens 1M dataset. Experimental results show that as ρ

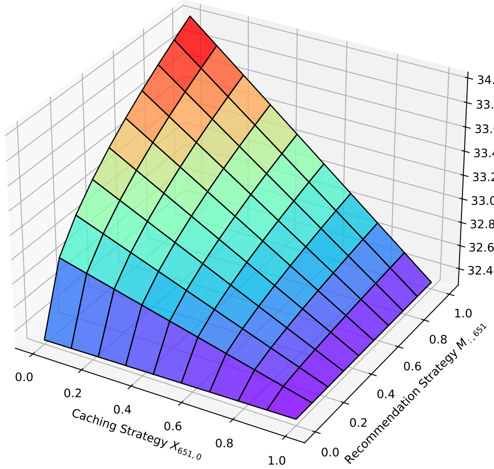

Fig. 4. Utility of the CP under the Stackelberg game formulation.

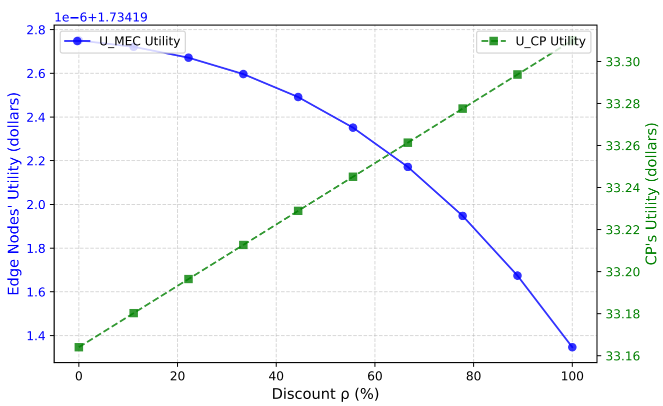

Fig. 5. The effect of discount ρ on the efficiency of CP and edge nodes.

increases, the utility of edge nodes decreases gradually. This is because a higher ρ incentivizes the CP to enforce direct content transmission from the CP when the edge node cannot find the requested content locally, thereby reducing the edge node's utility. In contrast, the utility of the CP increases linearly and significantly with higher ρ, indicating that although a larger discount boosts the CP's content transfer volume, the potential gains from content recommendations far outweigh the associated transmission costs. Based on these findings, we conclude that a 60% discount strikes a favorable balance for collaboration between the CP and edge nodes, ensuring that both parties benefit from the cooperation under the given system parameters.

VI. Conclusion

In this paper, we proposed GT-FMC, a federated multi-edge caching framework that integrates a two-stage game theoretic design. First, we introduced a distributed collaborative caching framework that reduces carbon emissions by intelligently coordinating edge nodes to optimize content decisions while simultaneously ensuring optimal integration of content providers, edge nodes, and users. Next, we developed a lightweight federated content popularity prediction method, leveraging TCN to collaboratively learn global content popularity across multiple edge nodes. We then investigated the joint optimization of utilities for all three parties, considering factors such as the user-to-edge node topology, recommendation strategy, and caching strategy. To solve the coupled non-linear optimization problem, we proposed a two-stage game-theoretic algorithm based on a matching game and the Stackelberg game. Through experiments conducted on real-world testbeds and datasets, we demonstrated that GT-FMC significantly outperforms baseline methods in terms of cache hit rate and energy efficiency. Furthermore, we validated the effectiveness of the proposed matching game and Stackelberg game models using matching maps and utility analyses, which support the rationality of the game-theoretic approach. Lastly, we provided insights into the cooperation discount between content providers and edge nodes, offering a deeper understanding of the underlying collaborative dynamics.

The proposed framework demonstrates strong potential for deployment in a variety of real-world IIoT scenarios. In advanced manufacturing environments, AR/VR terminals are increasingly utilized for immersive training, remote assistance, and digital quality inspection, all of which demand high data throughput and low latency. In large industrial campuses and ports, autonomous vehicles rely on timely access to traffic updates, HD maps, and entertainment systems to ensure safe and efficient operation. Additionally, digital twin systems, commonly used in wind farms and offshore oil rigs, require frequent synchronization of telemetry data from distributed sensors for real-time equipment monitoring and fault prediction. These scenarios underscore the practical value of edge-enabled, energy-aware content delivery mechanisms in supporting the next generation of intelligent industrial systems.

References

[1] S. Zhu, K. Ota, and M. Dong, "Green ai for iiot: Energy efficient intelligent edge computing for industrial internet of things," *IEEE Transactions on Green Communications and Networking*, vol. 6, no. 1, pp. 79–88, 2021.

[2] Z. Chen, S. Huang, G. Min, Z. Ning, J. Li, and Y. Zhang, "Mobility-aware seamless service migration and resource allocation in multi-edge iov systems," *IEEE Transactions on Mobile Computing*, vol. 24, no. 7, pp. 6315–6332, 2025.

[3] J. Chi, X. Zhou, F. Xiao, Y. Lim, and T. Qiu, "Task offloading via prioritized experience-based double dueling dqn in edge-assisted iiot," *IEEE Transactions on Mobile Computing*, 2024.

[4] G. Zheng, Q. Ni, K. Navaie, and H. Pervaiz, "Semantic communication in satellite-borne edge cloud network for computation offloading," *IEEE Journal on Selected Areas in Communications*, vol. 42, no. 5, pp. 1145–1158, 2024.

[5] P. Lin, Z. Ning, Z. Zhang, Y. Liu, F. R. Yu, and V. C. Leung, "Joint optimization of preference-aware caching and content migration in cost-efficient mobile edge networks," *IEEE Transactions on Wireless Communications*, 2023.

[6] T. Zong, C. Li, Y. Lei, G. Li, H. Cao, and Y. Liu, "Cocktail edge caching: Ride dynamic trends of content popularity with ensemble learning," *IEEE/ACM Transactions on Networking*, vol. 31, no. 1, pp. 208–219, 2022.

[7] C.-C. Lin, Y. Chiang, and H.-Y. Wei, “Collaborative edge caching with multiple virtual reality service providers using coalition games,” in *2023 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1–6, IEEE, 2023.
[8] D. Tsigkari, G. Iosifidis, and T. Spyropoulos, “Quid pro quo in streaming services: Algorithms for cooperative recommendations,” *IEEE Transactions on Mobile Computing*, vol. 23, no. 2, pp. 1753–1768, 2023.
[9] Z. Chen, J. Zhang, G. Min, Z. Ning, and J. Li, “Traffic-aware lightweight hierarchical offloading toward adaptive slicing-enabled sagin,” *IEEE Journal on Selected Areas in Communications*, vol. 42, no. 12, pp. 3536–3550, 2024.
[10] Z. Yu, J. Hu, G. Min, Z. Wang, W. Miao, and S. Li, “Privacy-preserving federated deep learning for cooperative hierarchical caching in fog computing,” *IEEE Internet of Things Journal*, vol. 9, no. 22, pp. 22246–22255, 2022.
[11] Z. Chen, J. Liang, Z. Yu, H. Cheng, G. Min, and J. Li, “Resilient collaborative caching for multi-edge systems with robust federated deep learning,” *IEEE Transactions on Networking*, pp. 1–16, 2024.
[12] H. Zhou, H. Wang, Z. Yu, G. Bin, M. Xiao, and J. Wu, “Federated distributed deep reinforcement learning for recommendation-enabled edge caching,” *IEEE Transactions on Services Computing*, vol. 17, no. 6, pp. 3640–3656, 2024.
[13] G. Zheng, Q. Ni, A. Kaushik, and L. Yang, “Semantic communication meets heterogeneous network: Emerging trends, opportunities, and challenges,” *arXiv preprint arXiv:2502.08999*, 2025.
[14] Z. Chen, B. Xiong, X. Chen, G. Min, and J. Li, “Joint computation offloading and resource allocation in multi-edge smart communities with personalized federated deep reinforcement learning,” *IEEE Transactions on Mobile Computing*, vol. 23, no. 12, pp. 11604–11619, 2024.
[15] P. Qin, Y. Fu, J. Zhang, S. Geng, J. Liu, and X. Zhao, “Drl-based resource allocation and trajectory planning for noma-enabled multi-uav collaborative caching 6g network,” *IEEE Transactions on Vehicular Technology*, vol. 73, no. 6, pp. 8750–8764, 2024.
[16] B. Liu, C. Liu, and M. Peng, “Dynamic cache placement and trajectory design for uav-assisted networks: A two-timescale deep reinforcement learning approach,” *IEEE Transactions on Vehicular Technology*, vol. 73, no. 4, pp. 5516–5530, 2024.
[17] Y. Fu, Y. Zhang, A. K. Y. Wong, and T. Q. S. Quek, “Revenue maximization: The interplay between personalized bundle recommendation and wireless content caching,” *IEEE Transactions on Mobile Computing*, vol. 22, no. 7, pp. 4253–4265, 2023.
[18] C. Sun, X. Li, J. Wen, X. Wang, Z. Han, and V. C. M. Leung, “Federated deep reinforcement learning for recommendation-enabled edge caching in mobile edge-cloud computing networks,” *IEEE Journal on Selected Areas in Communications*, vol. 41, no. 3, pp. 690–705, 2023.
[19] Y. Zhang, Y. Zhou, S. Zhang, G. Gui, B. Adebisi, H. Gacanin, and d. H. Sari, “An efficient caching and offloading resource allocation strategy in vehicular social networks,” *IEEE Transactions on Vehicular Technology*, vol. 73, no. 4, pp. 5690–5703, 2024.
[20] D. Chen, W. Xu, H. Wang, Y. Qi, R. Li, P. Zhou, and S. Guo, “Caching user-generated content in distributed autonomous networks via contextual bandit,” *IEEE Transactions on Mobile Computing*, vol. 23, no. 8, pp. 8355–8369, 2024.
[21] Q. Wu, Y. Zhao, Q. Fan, P. Fan, J. Wang, and C. Zhang, “Mobility-aware cooperative caching in vehicular edge computing based on asynchronous federated and deep reinforcement learning,” *IEEE Journal of Selected Topics in Signal Processing (JSTSP)*, vol. 17, no. 1, pp. 66–81, 2023.
[22] X. Kang and S. Sun, “Incentive mechanism design for mobile data offloading in heterogeneous networks,” in *2015 IEEE International Conference on Communications (ICC)*, pp. 7731–7736, 2015.
[23] G. Ma, Z. Wang, M. Zhang, J. Ye, M. Chen, and W. Zhu, “Understanding performance of edge content caching for mobile video streaming,” *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 5, pp. 1076–1089, 2017.
[24] Z. Chen, Q. Jiang, L. Chen, X. Chen, J. Li, and G. Min, “Mc-2pf: A multi-edge cooperative universal framework for load prediction with personalized federated deep learning,” *IEEE Transactions on Mobile Computing*, vol. 24, no. 6, pp. 5138–5154, 2025.
[25] X. Zhang, Z. Qi, G. Min, W. Miao, Q. Fan, and Z. Ma, “Cooperative edge caching based on temporal convolutional networks,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 33, no. 9, pp. 2093–2105, 2022.
[26] H. Lim, D. G. Andersen, and M. Kaminsky, “3lc: Lightweight and effective traffic compression for distributed machine learning,” *Proceedings of Machine Learning and Systems*, vol. 1, pp. 53–64, 2019.
[27] D. Gale and L. S. Shapley, “College admissions and the stability of marriage,” *The American Mathematical Monthly*, vol. 120, no. 5, pp. 386–391, 2013.
[28] F. M. Harper and J. A. Konstan, “The movielens datasets: History and context,” *Acm transactions on interactive intelligent systems (tiis)*, vol. 5, no. 4, pp. 1–19, 2015.
[29] L. Chen, S. Zhou, and J. Xu, “Energy efficient mobile edge computing in dense cellular networks,” in *2017 IEEE International Conference on Communications (ICC)*, pp. 1–6, IEEE, 2017.
[30] Y. Zhang, M. A. Kishk, and M.-S. Alouini, “Computation offloading and service caching in heterogeneous mec wireless networks,” *IEEE Transactions on Mobile Computing*, vol. 22, no. 6, pp. 3241–3256, 2021.
[31] J. Wang, N. Xie, V. Ilea, C. Bovo, H. Xin, and Y. Wang, “Cooperative trading mechanism and satisfaction-based profit distribution in a multi-energy community,” *Frontiers in energy research*, vol. 9, p. 723192, 2021.
[32] D. C. Nguyen, M. Ding, P. N. Pathirana, A. Seneviratne, J. Li, and H. V. Poor, “Federated learning for internet of things: A comprehensive survey,” *IEEE Communications Surveys & Tutorials*, vol. 23, no. 3, pp. 1622–1658, 2021.
[33] S. Müller, O. Atan, M. van der Schaar, and A. Klein, “Context-aware proactive content caching with service differentiation in wireless networks,” *IEEE Transactions on Wireless Communications (TWC)*, vol. 16, no. 2, pp. 1024–1036, 2016.
[34] M. Dehghan, L. Massoulié, D. Towsley, D. S. Menasché, and Y. C. Tay, “A utility optimization approach to network cache design,” *IEEE/ACM Transactions on Networking*, vol. 27, no. 3, pp. 1013–1027, 2019.
[35] G. Quan, J. Tan, and A. Eryilmaz, “Counterintuitive characteristics of optimal distributed lru caching over unreliable channels,” *IEEE/ACM Transactions on Networking (ToN)*, vol. 28, no. 6, pp. 2461–2474, 2020.
[36] Y. Chen, Y. Liu, J. Zhao, and Q. Zhu, “Mobile edge cache strategy based on neural collaborative filtering,” *IEEE Access*, vol. 8, pp. 18475–18482, 2020.

Jie Liang is working toward his M.S. degree in Computer Science with the College of Computer and Data Science at the Fuzhou University, China. He received his B.S. degree in Software Engineering from the Fujian University of Technology, China. His research interests include edge caching and federated learning.

Zhengxin Yu is currently a Lecturer with the School of Computing and Communications at the Lancaster University, UK. She received the Ph.D. degree in Computer Science from the University of Exeter, UK. Her research interests focus on Federated Learning, Deep Learning, Cyber Security, and Multi-access Edge Computing.

Haris Pervaiz received the M.Sc. degree in information security from the Royal Holloway University of London, Egham, U.K., in 2005, and the Ph.D. degree from the School of Computing and Communication, Lancaster University, Lancaster, U.K., in 2016. He was a Lecturer at the School of Computing and Communications, Lancaster University, between 2018 and 2023. He is an Associate Professor with the School of Computer Science and Electronic Engineering, University of Essex, U.K. His research interests include 5G and beyond, 5G & beyond, green communication, semantic communication, AI/ML based networking, mmWave communication, energy & backhaul energy efficiency and green HetNets.

Guhan Zheng received the B.Eng. degree in telecommunication engineering from Zhengzhou University, China, in 2020, and the Ph.D. degree in communication systems from Lancaster University, U.K., in 2024. He was a Research Assistant with the University of Sussex, U.K., in 2023. He is currently a Senior Research Associate with the School of Computing and Communications at Lancaster University. His research interests include non-terrestrial networks, vehicular networks, semantic communications, and network economics.

Neeraj Suri holds the Distinguished Professorship and Chair in Cybersecurity at Lancaster University (UK and Leipzig/Germany). He is also an Adjunct Professor in the Dept. of CS at the Univ. of Massachusetts at Amherst. Suri previously held the Chair Professorship on “Dependable Systems & Software” at TU Darmstadt, Germany. His research interests span the design, analysis and assessment of trustworthy (dependable & secure) Cloud systems and software.