

# PPVF: An Efficient Privacy-Preserving Online Video Fetching Framework with Correlated Differential Privacy

Xianzhi Zhang, Yipeng Zhou, Di Wu, Quan Z. Sheng, Miao Hu, and Linchang Xiao

**Abstract**—Online video streaming has evolved into an integral component of the contemporary Internet landscape. Yet, the disclosure of user requests presents formidable privacy challenges. As users stream their preferred online videos, their requests are automatically seized by video content providers, potentially leaking users’ privacy. Unfortunately, current protection methods are not well-suited to preserving user request privacy from content providers while maintaining high-quality online video services. To tackle this challenge, we introduce a novel Privacy-Preserving Video Fetching (PPVF) framework, which utilizes trusted edge devices to pre-fetch and cache videos, ensuring the privacy of users’ requests while optimizing the efficiency of edge caching. More specifically, we design PPVF with three core components: (1) *Online privacy budget scheduler*, which employs a theoretically guaranteed online algorithm to select non-requested videos as candidates with assigned privacy budgets. Alternative videos are chosen by an online algorithm that is theoretically guaranteed to consider both video utilities and available privacy budgets. (2) *Noisy video request generator*, which generates redundant video requests (in addition to original ones) utilizing correlated differential privacy to obfuscate request privacy. (3) *Online video utility predictor*, which leverages federated learning to collaboratively evaluate video utility in an online fashion, aiding in video selection in (1) and noise generation in (2). Finally, we conduct extensive experiments using real-world video request traces from Tencent Video. The results demonstrate that PPVF effectively safeguards user request privacy while upholding high video caching performance.

**Index Terms**—Request Privacy, Video Pre-Fetching, Edge Caching, Online Algorithm, Correlated Differential Privacy.

## I. INTRODUCTION

ONLINE video streaming has become an indispensable service in our daily lives, serving billions of Internet users by streaming diverse videos, including movies, news, and TV episodes. In 2023, YouTube alone provided over 5 billion online videos daily to more than 122 million Internet users, with a total daily playback time exceeding 1 billion hours [1]. To serve such a huge number of users, it is common to leverage edge devices (EDs) to pre-fetch video content for users. Such EDs, including user devices [2], [3], [4], mobile vehicles [5], [6], [7], access points (APs) [4], [8], and edge nodes of the

content delivery network (CDN) [9], can significantly reduce communications between users and video content providers (CPs). Meanwhile, serving users with content cached on EDs can achieve a high quality of service (QoS) and a low playback latency for video streaming.

However, with the proliferation of the online video market, the *request privacy leakage* concern has risen [7], [10], [11]. When users request videos from online CPs, their request traces are automatically recorded by CPs. Analyzing these traces can potentially reveal sensitive privacy information such as gender [9], age [12], location [2], [6], and hobbies [13], [14]. The leakage of request privacy poses significant risks to users with the spread of spam and scams [15]. Consequently, it is urgent and essential to develop effective video request strategies with preserved user privacy [16].

Various privacy-preserving methodologies, such as encryption, federated learning (FL) and differential privacy (DP), have emerged, but applying them to preserve video request privacy is non-trivial, which can be explained as follows. Encryption-based methods, e.g., Hypertext Transfer Protocol Secure (HTTPS) [17], can only shield against privacy threats from external attackers when delivering videos. FL is a generic framework to preserve privacy by only exposing parameters when training machine learning models for edge devices [18], [19]. Yet, they fail to conceal request traces from CPs because user requests must remain visible to CPs for video streaming services to function properly. DP is a widely utilized approach to safeguarding user privacy in machine learning systems [12], [20], [21]. However, straightly distorting user requests with DP noises can severely impair video streaming efficiency by requesting many videos out of users’ interest.

To mitigate video request privacy leakage, we propose a novel Privacy-Preserving Video Fetching (PPVF) framework by synthetically utilizing video pre-fetching, FL and correlated differential privacy (CDP) to supplement each other and overcome the deficiency of each single methodology. For deployment in practice, our framework can be implemented on trusted EDs, e.g., user devices [2], [3], [4], vehicles [5], [6], [7], access points (APs) [4], [8], owned or trusted by users. By utilizing the PPVF framework, trusted EDs achieve a harmonious blend of efficient video delivery and safeguard the privacy of viewing records. In summary, our main contributions are listed below:

- To the best of our knowledge, we are among the first to propose an *online privacy budget scheduler* for reconciling privacy and efficiency in online video systems. Can-

Xianzhi Zhang, Di Wu, Miao Hu, Linchang Xiao are with the School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou, 510006, China, and the Key Laboratory of Machine Intelligence and Advanced Computing (Sun Yat-sen University), Ministry of Education, China. E-mail: {zhangxzh9, xiaolch3}@mail2.sysu.edu.cn; {wudi27, hu-miao5}@mail.sysu.edu.cn. (Di Wu is the corresponding author.)

Yipeng Zhou, and Quan Z. Sheng are with the School of Computing, Macquarie University, NSW 2109, Australia. E-mail: {yipeng.zhou, michael.sheng}@mq.edu.au.

didate videos with assigned privacy budgets are tactfully selected by a threshold-based online algorithm for further distorting the process. Additionally, the performance of the allocation algorithm is theoretically guaranteed by competitive analysis.

- We leverage Correlated Differential Privacy (CDP) to design the *noisy video request generator* for generating video requests (including redundant noisy ones) in edge video caching systems. By taking correlated video request patterns into account, CDP can more accurately calibrate the noise scale when distorting pre-fetching requests and hence avoid injecting excessive noises.
- To predict video utility (serving as the pivotal prior knowledge for video selection and noise generation), we further construct a FL-based *online video utility predictor*, which only exposes non-critical parameters to collaboratively evaluate video utility in an online and privacy-preserving mode.
- We conduct extensive experiments by using real-world request traces collected from Tencent Video [22] to validate the superiority of PPVF. The experimental results demonstrate that PPVF is the best in preserving privacy without significantly compromising caching performance compared with the state-of-the-art baselines.

The remainder of the paper is organized as follows. The PPVF system architecture, the threat model, and the problems formulation are presented in Section II. In Section III, we introduce novel algorithms for allocating the privacy budget and determining the pre-fetching strategy in edge caching systems. Section IV presents the video utility predictor obtained via federated learning and online parameter estimation methods. The experimental results are reported in Section V followed by a discussion of the related works in Section VI. Finally, we conclude our paper in Section VII.

## II. SYSTEM MODEL AND PRELIMINARY

In this section, we introduce the system model of our PPVF framework and the main entities in PPVF. To facilitate readability, we have summarized notations in Table I.

### A. System Architecture

There are three types of entities in the system, which are briefly introduced as follows:

- **Content Provider (CP):** The CP is an online video service provider possessing a comprehensive set of  $I$  videos denoted by  $\mathcal{I} = \{i_1, i_2, \dots, i_I\}$ . However, the CP also collects users' request traces to enhance its services, e.g., recommendation [23], [13] and advertisement [12], [14], which is regarded as the main risk entity in our privacy model.
- **Trusted Edge Devices (EDs):** In PPVF, we denote  $\mathcal{E} = \{e_1, e_2, \dots, e_E\}$  as the set of all EDs which can be trusted by users [24], [10], [11]. Each ED  $e \in \mathcal{E}$  has a storage limitation  $c_e$  in our system model. EDs play two critical roles: i) caching videos fetched from the CP to serve users' requests with a higher quality of service (QoS),

and ii) preserving video request privacy to conceal users' real preferences<sup>1</sup>.

- **Users:** Users are final consumers of videos. Rather than directly exposing video requests to the CP, users in PPVF only submit their requests to corresponding trusted EDs, which act as agents to fetch videos from the CP for users.

**1) Interactions Between EDs and Users:** When a user needs to watch video content  $i' \in \mathcal{I}$  at time  $\tau \in [0, T)$ , the user submits her request (denoted by a view content vector  $\mathbf{v}_e = [v_{e,i}]^1$  where  $v_{e,i'} = 1$  and  $v_{e,i} = 0, \forall i \neq i', i \in \mathcal{I}$ ) to its ED  $e$ . Here, time interval  $[0, T)$  is the observed window in our problem. If the requested video is cached at ED  $e$ , the video can be streamed directly from the ED to the user. Otherwise, ED  $e$  needs to pre-fetch some redundant videos plus the missing video  $i'$  from the CP. Here, we denote  $k \in \mathbb{N}^+, 1 \leq k \leq K_e$  as the index of the cache missing request that needs to be fetched from the remote CP by ED  $e$ , where  $K_e$  represents the maximum index number of the missing requests at ED  $e$  in  $[0, T)$ .

Note that all EDs can authentically record all video requests from users to evaluate the utility for video pre-fetching and caching [24], [4], [12]. For any ED  $e$ , let  $\mathcal{V}_e^k = \{(i, \tau) \mid i \in \mathcal{I}, \tau \in [0, t^k), \tau \in \mathbb{R}, 1 \leq k \leq K_e\}$  denote the set of all historical viewing requests up to the time  $t^k$ , where  $t^k$  is the time of the  $k$ -th cache missing request. Besides, we denote  $\mathcal{T}_{e,i}^{t_1, t_2}$  as the timestamp set of the viewing requests for video  $i$  within time window  $[t_1, t_2)$  at ED  $e$ .

**2) Interactions between the CP and EDs:** In PPVF, EDs, in lieu of end users, interact with the CP. On the one hand, the CP delivers requested videos to EDs. On the other hand, since each ED has a limited number of local viewing requests, the CP needs to assist EDs in evaluating video utility with federated learning to enhance the quality of service.

The vector of pre-fetching requests is denoted by  $\mathbf{x}_e^k = [x_{e,i}^k]^1$ , where  $x_{e,i}^k \in \{0, 1\}, \forall e, i, k$ . To preserve privacy, ED  $e$  utilizes  $\mathbf{x}_e^k$  to obfuscate the original view request vector  $\mathbf{v}_e^k$  for the  $k$ -th cache missing request, which needs to be fetched from the CP. Videos finally fetched by ED  $e$  from the CP is conducted by  $\mathbf{r}_e^k = [r_{e,i}^k]^1$ , where  $r_{e,i}^k = v_{e,i}^k | x_{e,i}^k$ , representing the fetching vector sent by ED  $e$  for the  $k$ -th cache missing request. Here, the symbol ' $|$ ' represents the 'OR' operator, indicating that whether ED  $e$  fetches video  $i$  depends on both  $x_{e,i}^k$  and  $v_{e,i}^k$ . Note that the generation of  $\mathbf{v}_e^k$  is purely based on users' view interests, not affected by our strategies. Our study focuses on generating  $\mathbf{x}_e^k$  for privacy protection.

Furthermore, with the assistance of the CP, we assume that EDs can periodically update the parameters of their local model for video utility prediction without disclosing their private data. The set of time points to execute the online parameter estimation is denoted by  $t_\theta \in \mathcal{Q}$ , where  $\mathcal{Q} = \{t_\theta \mid t_\theta \in [0, T), t_\theta \in \mathbb{N}\}$ . Note that  $t_\theta$  represents the time point to update model parameters  $\theta$ , not the time point for requesting videos. Due to the limited caching space, ED  $e$  updates its cached videos by fetching videos according to

<sup>1</sup>Our main focus is to design a privacy framework for trusted edge devices, which may deploy Trusted Execution Environments (TEE), such as Intel SGX, TrustZone for Cortex-M, to convince users and execute instructions reliably.

TABLE I: Main notations used in the paper.

Notation	Description
$\mathcal{E} / \mathcal{I}$	The space set of all edge devices (EDs) / videos.
$i / e$	The index of any video / ED.
$k$	The index of request that is missed at a specific ED and needs to be fetched from the CP.
$t^k$	The timestamp of the $k$ -th cache missing request.
$\mathcal{V}_e / \mathcal{V}_e^k$	The set of all viewing requests at ED $e$ in time $[0, T) / [0, t^k)$ .
$\mathcal{T}_{e,i} / \mathcal{T}_{e,i}^{t_1,t_2}$	The timestamp set of the viewing requests of video $i$ arriving at ED $e$ in time window $[0, T) / [t_1, t_2)$ .
$\mathbf{x}_e^k / \mathbf{v}_e^k$	The pre-fetching / viewing vector for the $k$ -th cache missing request at ED $e$ .
$c_e / f_e$	The caching / pre-fetching capability of ED $e$ .
$\alpha_e^k$	The privacy budget allocation vector for the $k$ -th pre-fetching at ED $e$ .
$\mathcal{A}_e^k$	The candidate video set for generating redundant requests for the $k$ -th pre-fetching at ED $e$ .
$\xi_{e,i} / \epsilon_{e,i}$	The total privacy budget / once privacy cost of ED $e$ with respect to video $i$ .
$\lambda_{e,i}^k$	The utility of pre-fetching and caching video $i$ in time $t^k$ at ED $e$ .
$\theta = \{\beta, \mathbf{p}, \mathbf{q}\}$	The parameters of MEP model.
$t_\theta$	The update time point of online parameter estimation based on FL-framework.
$\Psi_e^k$	The correlated degree matrix among different videos in time $t^k$ at ED $e$ .
$\psi_e^k / \alpha_e^k / \sigma_e^k$	The historical matrices to calculate the correlated degree matrix $\Psi_e^k$ in time $t^k$ at ED $e$ .
$\Delta\lambda_{e,i}^k / \Delta\lambda_{e,gc}^k$	The correlated sensitivity for each video $i$ / global of the $k$ -th pre-fetching at ED $e$ .

predicted video utility when its cache space is full. The video utility will be further specified in Section IV.

### B. Threat Model

In traditional online video systems, privacy threats related to video fetching primarily arise from the exposure of users' video-request patterns and preferences. As users interact with the CP to access the online video services, their historical video requests and pre-fetching activities will be inadvertently exposed to the CP, which can accordingly infer sensitive information, such as age [12], gender [9], locations [2], [6], and favorites [13], [14], of users. Such threats are driven by the goal of enhancing services through caching or recommendation algorithms. CPs can exploit inferred sensitive information to gain insights into individual user preferences. Therefore, unauthorized access to user-specific information without protection poses a significant privacy threat, enabling CPs to infer personal preferences, potentially compromising users' privacy.

### C. Problems Formulation

Let us first consider the global video caching problem without considering privacy leakage. When requesting videos missed by the edge cache from the CP, ED  $e$  also makes requests for redundant videos based on pre-fetching decisions  $\mathbf{x}_e = [x_{e,i}^k]^{K_e \times I}$ . Let  $\lambda_e = [\lambda_{e,i}^k]^{K_e \times I}$  denote all video utility values, e.g., the predicted rate to request videos by users, for any ED  $e$ . The problem of maximizing pre-fetching and caching utility can be formulated by:

$$\mathbb{P}_g : \max_{\mathbf{x}_e, \forall e} \sum_{e \in \mathcal{E}} \sum_{k=1}^{K_e} \sum_{i \in \mathcal{I}} \lambda_{e,i}^k \cdot x_{e,i}^k \quad (1a)$$

$$\text{s.t.} \sum_{i \in \mathcal{I}} x_{e,i}^k \leq f_e, \quad \forall e \in \mathcal{E}, 1 \leq k \leq K_e, \quad (1b)$$

$$x_{e,i}^k \in \{0, 1\}, \quad \forall e \in \mathcal{E}, \forall i \in \mathcal{I}, 1 \leq k \leq K_e, \quad (1c)$$

$$\lambda_{e,i}^k = h_e(i, t^k \mid \mathcal{V}_e^k, \theta), \forall e \in \mathcal{E}, \forall i \in \mathcal{I}, 1 \leq k \leq K_e, \quad (1d)$$

where  $h_e : \mathcal{I} \times \mathbb{R}^+ \rightarrow \mathbb{R}^+$  can represent any prediction function for utility with model parameters  $\theta$  and historical viewing records  $\mathcal{V}_e^k$  up to time  $t^k$  at ED  $e$ . Besides, Eq. (1b) restricts the maximum pre-fetching capacity  $f_e$  of ED  $e$ .

For traditional video streaming, the CP can collect historical video request records to infer  $\lambda_e, \forall e$ , which can be further used to derive optimal solution  $\mathbf{x}_e^*$ , for all EDs. In this process, the CP can exactly infer preferences exposed by EDs. To prevent privacy leakage, EDs can apply differential privacy (DP) noises to distort pre-fetching decisions  $\mathbf{x}_e$ , to hide both users' original video requests and video utility. It is difficult for the CP to infer user privacy from public fetching actions, and hence user privacy is preserved. In the rest of the subsection, we extend  $\mathbb{P}_g$  to present the privacy-preserving video pre-fetching problem.

We begin by succinctly introducing DP, avoiding any unnecessary notation. In problem  $\mathbb{P}_g$ , the pre-fetching decision variables  $\mathbf{x}$  are primarily determined by the utility  $\lambda$ , which is evaluated by the function  $h$  with the parameter  $\theta$ , and the set  $\mathcal{V}$  of real request records. To preserve privacy, DP can be applied to distort the output of utility function  $h$  to protect privacy in dataset  $\mathcal{V}$ .

**Definition 1. ( $\epsilon$ -Differential Privacy)** A randomized mechanism  $\mathcal{M}$  confirms  $\epsilon$ -DP, if for any pair of adjacent datasets  $\mathcal{V} \simeq \mathcal{V}'$ , any tuple of input  $(i, t) \in \mathcal{I} \times \mathbb{R}^+$ , and any predict function  $h$  with its parameters  $\theta$ , it satisfies:

$$\frac{\Pr\{\mathcal{M}(h(i, t \mid \mathcal{V}, \theta)) \in \mathcal{O}\}}{\Pr\{\mathcal{M}(h(i, t \mid \mathcal{V}', \theta)) \in \mathcal{O}\}} \leq \exp(\epsilon). \quad (2)$$

Here,  $\epsilon$  is the privacy budget and  $\mathcal{O}$  represents the outcome range of mechanism  $\mathcal{M}$ .

However, in practical online video systems, cardinality  $I$  for  $\mathcal{I}$  is a huge number, and users' view preferences can be very skewed, implying that there exists a large number of cold videos with very few user requests [25], [4]. Thereby, directly applying DP noise to distort utilities for video pre-fetching confronts the following two challenges:

- (1) More privacy budget will be consumed to protect privacy if there are more videos in  $\mathcal{I}$ . If cardinality  $I$  is a large

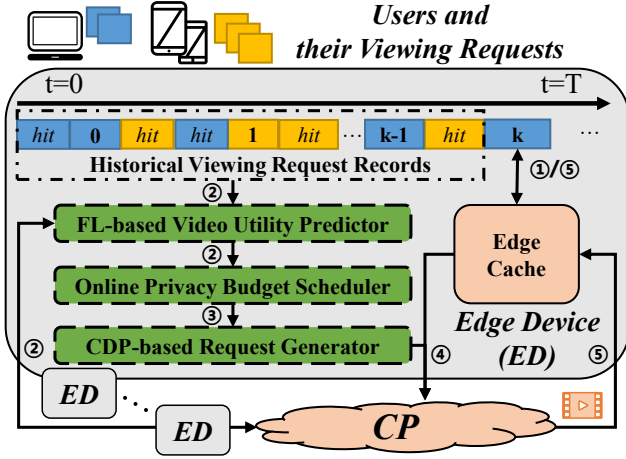


Fig. 1: The workflow of privacy-preserving video fetching (PPVF) for online video service at EDs.

number, the noises will be excessively large such that the video utility predicted by the function  $h$  is valueless.

- (2) Considering that the CP can implement collaborative filtering algorithms to infer user privacy, requesting cold videos (with scarce requests) as noises is not effective in preserving privacy since collaborative filtering algorithms can easily remove the noisy influence of cold video requests [26].

To tackle these two challenges, we consider adding DP noises to protect pre-fetching privacy by the Exponential Mechanism (EM) [27] with a candidate video set selected in an online manner. We start with a brief introduction to the EM. The EM is a classical DP mechanism satisfying  $\epsilon$ -DP, which can be applied to distort the output of utility function  $h$ , defined as follows.

**Definition 2.** (Exponential Mechanism) The exponential mechanism (EM) satisfies  $\epsilon$ -differential privacy with the following steps: (1) specifies a global sensitivity, denoted as  $\Delta\lambda$ , for a video utility prediction function  $h : \mathcal{I} \times \mathbb{R}^+ \rightarrow \mathbb{R}^+$ . (2) video  $i \in \mathcal{I}$  is selected to request with the probability

$$P\{i\} \propto \exp\left(\frac{\epsilon \cdot h(i, t \mid \mathcal{V}, \theta)}{2\Delta\lambda}\right).$$

Here,  $\epsilon$  represents the privacy budget,  $\mathcal{V}$  is the set of request records privately owned by an ED, and  $\theta$  represents parameters in function  $h$ .

Instead of applying the whole video set  $\mathcal{I}$ , we identify a candidate video set  $\mathcal{A}_e^k \subseteq \mathcal{I}$  for ED  $e$  to generate redundant requests ED  $e$ . Here,  $k$  is the pre-fetching index for the request missed by the edge cache. The whole candidate video set  $\mathcal{A}_e = \{\mathcal{A}_e^k \mid 1 \leq k \leq K_e\}$  is obtained by solving the problem  $\mathbb{P}_e$  subjecting to both the privacy budget and pre-fetching capacity constraint. In the problem  $\mathbb{P}_e$ , the constraint (3b) ensures that the assigned privacy budget cannot exceed the total privacy budget  $\xi_e$  for each video, and  $f_e$  denotes the

pre-fetching capability at ED  $e$ .

$$\mathbb{P}_e : \mathcal{J}_e^* = \max_{\mathbf{a}_e} \sum_{k=1}^{K_e} \sum_{i \in \mathcal{I}} \lambda_{e,i}^k \cdot a_{e,i}^k \quad (3a)$$

$$\text{s.t.} \sum_{k=1}^{K_e} \epsilon_{e,i} \cdot a_{e,i}^k \leq \xi_e, \quad \forall i \in \mathcal{I}, \quad (3b)$$

$$\sum_{i \in \mathcal{I}} a_{e,i}^k \leq f_e, \quad 1 \leq k \leq K_e, \quad (3c)$$

$$a_{e,i}^k \in \{0, 1\}, \quad \forall i \in \mathcal{I}, 1 \leq k \leq K_e, \quad (3d)$$

$$\lambda_{e,i}^k = h_e(i, t^k \mid \mathcal{V}_e^k, \theta), \quad \forall i \in \mathcal{I}, 1 \leq k \leq K_e, \quad (3e)$$

$$\mathcal{A}_e^k = \{i \mid a_{e,i}^k = 1, \forall i \in \mathcal{I}\}, \quad 1 \leq k \leq K_e. \quad (3f)$$

Based on problem  $\mathbb{P}_e$ , we can illustrate the holistic optimization process of PPVF:

- **Federated Learning:** EDs can jointly evaluate video utility, i.e.,  $\lambda_{e,i}^k = h_e(i, t^k \mid \mathcal{V}_e^k, \theta)$ ,  $\forall e, i, k$ , via the federated learning framework. This approach allows EDs to achieve significantly more accurate video utility than those obtained solely from local traces.
- **Video Selection and Budget Allocation:** With evaluated  $\lambda_e^k$ , EDs can solve  $\mathbb{P}_e$  in an online manner to obtain  $\mathcal{A}_e^k$ , which is the candidate set of videos to be requested from the remote CP.
- **Pre-fetching Request Generation:** With  $\mathcal{A}_e^k$  derived from  $\mathbb{P}_e$  and privacy budget allocation decisions, EDs can apply the EM to distort their fetching requests with correlated differential privacy (CDP). Then, EDs contact the CP to fetch both viewing videos and redundant videos.

### III. PPVF FRAMEWORK DESIGN

#### A. PPVF Overview

To better understand how users, EDs and the CP interact with each other, we present the workflow of PPVF, as shown in Fig. 1. Briefly speaking, the life cycle of each request involves five steps. ① Upon receiving a video viewing request from a user, EDs first search for that video in their *edge cache*. If the video is cached, EDs directly stream it to users with a shorter response latency without any privacy leakage. Otherwise, EDs assemble pre-fetching video requests, along with the viewing video, to fetch redundant videos from the CP. ② The *utility predictor* evaluates the videos' utility based on the federated learning framework, which ensures that only model parameters  $\theta$  are exchanged between EDs and the CP. The process is detailed in Section IV. ③ Subsequently, the *budget scheduler* assesses video utility in conjunction with the privacy budget to curate a video candidate set for the subsequent pre-fetching decision, elaborated in Section III-B. ④ The *request generator* distorts the original utility to generate a pre-fetching decision, which navigates the trade-off between privacy and caching performance by leveraging the EM with the CDP method. After combining the pre-fetching decision with the real view video, the final fetching vector is sent to the CP. This process will be discussed in Section III-C. ⑤ When the CP returns videos, EDs perform cache replacement based on video utility and forward the viewing video to users.

Note that our main contribution is represented by three modules in green color (long dashed box) in Fig. 1, which will be introduced in the following sections in detail.

### B. Threshold-based Online Algorithm

Directly solving problem  $\mathbb{P}_e$  confronts two challenges: (1) the problem is inherently online due to the dynamic nature of request patterns and video utility, and (2) the problem  $\mathbb{P}_e$  can be categorized as an online multiple knapsack problem, which is challenging to solve immediately and irrevocably, even if the utility  $\lambda_e^k$  is known.

To solve the challenging online problem  $\mathbb{P}_e$ , we propose a filtering mechanism [28], [29] that selects a video into  $\mathcal{A}_e^k$  if the ratio of this video utility over its excepted privacy budget exceeds a threshold. Intuitively speaking, a video of a higher utility will be played by users in the future with a higher probability. Hence, the utility of EDs in caching such videos will be higher. Meanwhile, considering the limited privacy budget, PPVF only selects videos with the ratio  $\lambda_{e,i}^k/\epsilon_{e,i}$  exceeding a threshold. This threshold is set in accordance with the fraction of the consumed privacy budget.

We can set the threshold for selecting videos as follows. Let  $U_e$  and  $L_e$  denote the upper and lower bound of the ratio for any video  $i$  at ED  $e$ , which means that  $L_e < \lambda_{e,i}^k/\epsilon_{e,i} < U_e$ ,  $\forall i \in \mathcal{I}, 1 \leq k \leq K_e$ . With the values of  $L_e$  and  $U_e$ , the threshold function is defined as:

$$\Theta_e(\gamma) = \begin{cases} \frac{L_e}{\left(\frac{U_e \cdot \exp(1)}{L_e}\right)^\gamma \cdot \frac{L_e}{e}}, & 0 \leq \gamma \leq \Gamma_e, \\ \Gamma_e < \gamma \leq 1. \end{cases} \quad (4)$$

Here,  $\gamma \in [0, 1]$  denotes the fraction of the privacy budget that has been allocated to a video until the current time,  $\Gamma_e = \frac{1}{1 + \ln(U_e/L_e)}$  is the lowest threshold for assessing the privacy budget proportion  $\gamma$ . The intuition of our design is that the selection of a video is more conservative if the consumed privacy budget fraction  $\gamma$  of that video is larger.

The detailed algorithm is presented in Alg. 1. Specifically, using the threshold function  $\Theta_e(\cdot)$ , PPVF randomly selects a video from the set  $\mathcal{I}$  and checks whether its  $\lambda_{e,i}^k/\epsilon_{e,i}$  exceeds the threshold  $\Theta_e(\gamma_{e,i}^k)$ . If so, the video is incorporated into candidate set  $\mathcal{A}_e^k$ ; if not, the video remains unchosen. This stochastic selection process continues until  $\mathcal{A}_e^k$  contains  $f_e$  videos. Although Alg. 1 is a heuristic-based algorithm, we can theoretically prove that Alg. 1 can achieve an optimal **competitive ratio** (CR) of  $(1 + \ln(U_e/L_e))$  for any ED  $e$ .

**Theorem 1.** *Alg. 1 has a competitive ratio of  $(1 + \ln(U_e/L_e))$  under rational Assumption 1 for any ED  $e$  to allocate the privacy budget.*

**Assumption 1.** *Each privacy cost of a pre-fetching video  $i$  has a weight much smaller than the total budget of the content, i.e.,  $\epsilon_{e,i} \ll \xi_e$  with respect to any ED  $e$ .*

*Proof.* We prove Theorem 1 with Assumption 1 in Appendix A.  $\square$

Considering  $U_e = \max_{\forall i,k} \lambda_{e,i}^k/\epsilon_{e,i}$  and  $L_e = \min_{\forall i,k} \lambda_{e,i}^k/\epsilon_{e,i}$ , we observe that CR is solely dependent on  $\lambda$  and  $\epsilon$  and is independent of the request quantity (i.e.,

---

**Algorithm 1:** Online privacy budget allocation algorithm for ED  $e$ .

---

**Input:** The space of all videos  $\mathcal{I}$ ; The total privacy budget for all videos  $\xi_e$ ; The pre-fetching capacity  $f_e$ ; The privacy cost  $\epsilon_e$ .

**Output:** The candidate set  $\mathcal{A}_e$  for video pre-fetching.

---

```

1 Initialize  $k \leftarrow 1, \gamma_e^k \leftarrow [0]^1$ ;
2 for  $k \leq K_e$  do
3   Initialize  $\mathbf{a}_e^k \leftarrow [0]^1, f^k \leftarrow 0, \mathcal{I}^k \leftarrow \mathcal{I}$ ;
4   Obtain the evaluated video utility  $\lambda_e^k$ ;
5   while  $f^k < f_e$  and  $\mathcal{I}^k \neq \emptyset$  do
6     Select content  $i$  randomly from  $\mathcal{I}^k$ ;
7      $\mathcal{I}^k \leftarrow \mathcal{I}^k - \{i\}$ ;
8     if  $\frac{\lambda_{e,i}^k}{\epsilon_{e,i}} > \Theta(\gamma_{e,i}^k)$  and  $\epsilon_{e,i} < (1 - \gamma_{e,i}^k) \cdot \xi_e$  then
9        $a_{e,i}^k \leftarrow 1; \gamma_{e,i}^{k+1} \leftarrow \gamma_{e,i}^k + \frac{\epsilon_{e,i}}{\xi_e}; f^k \leftarrow f^k + 1$ ;
10    else
11       $a_{e,i}^k \leftarrow 0; \gamma_{e,i}^{k+1} \leftarrow \gamma_{e,i}^k; f^k \leftarrow f^k$ ;
12    end
13  end
14  Generate the candidate set  $\mathcal{A}_e^k$  with  $\mathbf{a}_e^k$  following Eq. (3f);
15   $k \leftarrow k + 1$ ;
16 end
```

---

$K_e$ ). As the privacy budgets  $\epsilon$  are specified by EDs for each video, while utility  $\lambda$  is often generated by upstream utility prediction algorithms, they are both within a specific range. This characteristic ensures a constant-level CR of our algorithm, independent of the total request quantity (i.e.,  $K_e$ ), which is a highly appealing property. Through differentiation of CR with respect to  $U_e$  and  $L_e$ , it becomes evident that a decrease in  $U_e$  leads to a reduction in CR, indicating an approach to the optimal offline solution in the worst-case scenario. Similarly, a larger  $L_e$  results in a smaller value of CR. Moreover, when  $U_e/L_e$  approaches 1, CR approaches 1, indicating that the performance is close to the offline optimal solution.

### C. CDP-based Video Pre-fetching

Directly requesting videos in  $\mathcal{A}_e^k = \{i \mid a_{e,i}^k = 1, \forall i \in \mathcal{I}\}$  according to video utility can expose the video utility knowledge to the CP. To preserve privacy, PPVF adopts the EM to randomly select videos based on probability shown in Eq. (5) and generate the final pre-fetching decision  $\mathbf{x}_e^k$ . Specifically, if video  $i$  is selected by the EM, PPVF will set  $x_{e,i}^k = 1$  to pre-fetch that video from the CP. Otherwise, it will be set to 0. The probability is given by

$$P\{\text{video } i \text{ is chosen from } \mathcal{A}_e^k \mid \lambda_e^k\} \propto \exp \frac{\epsilon_e^k \cdot \lambda_{e,i}^k}{2 \cdot \Delta \lambda_{e,gc}^k}. \quad (5)$$

Here,  $\epsilon_e^k = \frac{1}{f_e} \sum_{i \in \mathcal{A}_e^k} \epsilon_{e,i}$  is the averaged privacy for pre-fetching one redundant video, where  $\sum_{i \in \mathcal{A}_e^k} \epsilon_{e,i}$  denotes the total privacy budget assigned by Alg. 1. Besides,  $\Delta \lambda_{e,gc}^k$  is the global sensitivity at the time of the  $k$ -th pre-fetching.

In our problem, the calculation of  $\Delta\lambda_{e,gc}^k$  is complicated because of the correlation between videos. Collaborative filtering algorithms can exploit such correlation information for inferring users' personal interests. To factor in the influence of video correlation, we employ the correlated differential privacy (CDP) for computing sensitivity. For ED  $e$ , we can calculate the correlation between videos  $i$  and  $j$  for the  $k$ -th pre-fetching with  $\lambda_{e,i(j)}^k$  predicted by utility function  $h_e$ . Suppose that EDs cache three history matrices  $\psi_e^{k-1} = [\psi_{e,i,j}^{k-1}]^{I \times I}$ ,  $\alpha_e^{k-1} = [\alpha_{e,i}^{k-1}]^I$ ,  $\sigma_e^{k-1} = [\sigma_{e,i}^{k-1}]^I$ , where the items can be incrementally updated by  $\psi_{e,i,j}^k = \psi_{e,i,j}^{k-1} + \lambda_{e,i}^k \cdot \lambda_{e,j}^k$ ,  $\alpha_{e,i}^k = \alpha_{e,i}^{k-1} + \lambda_{e,i}^k$ ,  $\sigma_{e,i}^k = \sigma_{e,i}^{k-1} + (\lambda_{e,i}^k)^2$ , respectively. Based on Pearson's correlation [27], the correlation degree between videos  $i$  and  $j$  can be calculated with these three history matrices as follows:

$$\Psi_{e,i,j}^k = \frac{k \cdot \psi_{e,i,j}^k - \alpha_{e,i}^k \cdot \alpha_{e,j}^k}{\sqrt{k \cdot \sigma_{e,i}^k - (\alpha_{e,i}^k)^2} \cdot \sqrt{k \cdot \sigma_{e,j}^k - (\alpha_{e,j}^k)^2}}. \quad (6)$$

Let  $\Psi_e^k = [\Psi_{e,i,j}^k]^{I \times I}$  denote the correlation matrix. The sensitivity of our problem can be calculated using the following two definitions.

**Definition 3.** (Correlated Video Sensitivity) For any given ED  $e \in \mathcal{E}$ , missing request index  $1 \leq k \leq K_e$  and video  $i, j \in \mathcal{A}_e^k$ , the correlated video sensitivity  $\Delta\lambda_{e,i}^k$  is defined as

$$\Delta\lambda_{e,i}^k = \sum_{j \in \mathcal{A}_e^k} (\Psi_{e,i,j}^k \|h_e(i, t^k | \mathcal{V}_e^k, \theta) - h_e(i, t^k | \mathcal{V}_{e,-j}^k, \theta)\|_1), \quad (7)$$

where  $\Psi_{e,i,j}^k$  is the correlation degree parameter,  $h_e: \mathcal{I} \times \mathbb{R}^+ \rightarrow \mathbb{R}^+$  is the utility function,  $\mathcal{A}_e^k$  is the candidate for video selection, and  $\mathcal{V}_e^k, \mathcal{V}_{e,-j}^k$  are two adjacent datasets different in video  $j$ .

Here, the L1 distance measures the effect on utility when deleting records related to video  $j$  from  $\mathcal{V}_e^k$ . Parameter  $\Psi_{e,i,j}^k$  estimates the correlated degree between videos  $i$  and  $j$ . Correlated Video Sensitivity combines the effect of correlated records and the correlated degree together.

**Definition 4.** (Correlated Sensitivity [27]) Given all the video sensitivities  $\Delta\lambda_{e,i}^k, \forall i \in \mathcal{A}_e^k$ , the global sensitivity  $\Delta\lambda_{e,gc}^k$  for the correlated videos is determined by

$$\Delta\lambda_{e,gc}^k = \max_{i \in \mathcal{A}_e^k} \Delta\lambda_{e,i}^k. \quad (8)$$

The correlated sensitivity lists all videos in candidate set  $\mathcal{A}_e^k$  responding to utility and selects the maximal video sensitivity as the correlated sensitivity. When videos are independent or weakly correlated, the global sensitivity will only slightly increase. Particularly, if all videos are independent, the correlated video sensitivity is equal to the global sensitivity, i.e.,  $\max_{i \in \mathcal{A}_e^k} \|h_e(i, t^k | \mathcal{V}_e^k, \theta) - h_e(i, t^k | \mathcal{V}_{e,-i}^k, \theta)\|_1$ . Additionally, given the historical matrices  $\psi_e^k, \alpha_e^k$ , and  $\sigma_e^k$ , which can be incrementally updated prior to the  $k$ -th pre-fetching, the correlated degree  $\Psi_{e,i,j}^k$  can be efficiently determined in  $O(1)$ , as per Eq. (6). Consequently, the computational complexity to obtain sensitivity  $\Delta\lambda_{e,i}^k$  of video  $i$  is  $O(f_e)$  with the evaluated utility vector  $\lambda_e^k$  by function  $h_e$  and the correlation matrix  $\Psi_e^k$ .

---

**Algorithm 2:** Online privacy-preserving videos pre-fetching algorithm for ED  $e$ .

---

**Input:** Pre-fetching index  $k$ ; Video utility  $\lambda_e^k$ ; Allocated privacy budget  $\epsilon_e$ ; Pre-fetching capacity  $f_e$ .

**Output:** The pre-fetching decision  $x_e$ .

- 1 Obtain candidate set  $\mathcal{A}_e^k$  for the  $k$ -th pre-fetching by Alg. 1;
  - 2 Incrementally update  $\psi_e^k, \alpha_e^k$ , and  $\sigma_e^k$  with video utility  $\lambda_e^k$ ;
  - 3 Calculate  $\Psi_e^k$  with matrices  $\psi_e^k, \alpha_e^k$ , and  $\sigma_e^k$  by Eq. (6);
  - 4 Obtain  $\Delta\lambda_{e,gc}^k$  with  $\lambda_e^k, \Psi_e^k$  and  $\mathcal{A}_e^k$  following Eqs. (7)-(8);
  - 5  $x_e^k \leftarrow [0]^I, f^k \leftarrow 0$ ;
  - 6 **while**  $f^k < f_e$  **do**
  - 7     Select pre-fetching video  $i$  from  $\mathcal{A}_e^k$  based on the probability shown in Eq. (5);
  - 8      $x_{e,i}^k = 1, f^k \leftarrow f^k + 1$ ;
  - 9 **end**
  - 10 **return** Pre-fetching decision  $x_e^k$ .
- 

Finally, the total computational complexity for deriving the correlated sensitivity  $\Delta\lambda_{e,gc}^k$  is  $O(f_e^2)$  with the utility vector  $\lambda_e^k$  and the correlation matrix  $\Psi_e^k$ .

The detailed algorithm to generate redundant pre-fetching video requests is presented in Alg. 2. It can be proved that Alg. 2 guarantees a  $\sum_{i \in \mathcal{A}_e^k} \epsilon_{e,i}$ -DP for the  $k$ -th pre-fetching decision at ED  $e$ , where  $\sum_{i \in \mathcal{A}_e^k} \epsilon_{e,i}$  denotes the total privacy budget assigned by Alg. 1. The proof can be directly deduced by considering the properties of the EM and the Composition Theorem [30]. The EM facilitates the selection of one video from the candidate set for pre-fetching while ensuring  $\epsilon_e^k$ -DP compliance. Based on the Composition Theorem [30], Alg. 1 makes a maximum  $f_e$  times of selections, adhering to  $f_e \cdot \epsilon_e^k$ -DP, which is equivalent to  $\sum_{i \in \mathcal{A}_e^k} \epsilon_{e,i}$ -DP.

**Remark 1.** In a nutshell, the superiority of PPVF for optimally balancing privacy preservation and efficiency is attributed to the following two advantages. Firstly, rather than blindly distorting requests for all videos, PPVF can reduce the consumption of the privacy budget by only distorting requests for a subset of selected candidate videos. Secondly, by considering correlation in video requests, PPVF can more accurately calibrate the noise scale by using CDP, which can avoid setting over-large noise scales for videos.

#### IV. ONLINE VIDEO UTILITY PREDICTION

In this section, we shift our focus to the discussion on evaluating video utility, i.e.,  $\lambda_{e,i}^k = h_e(i, t^k | \mathcal{V}_e^k, \theta), \forall e, i, k$ , by federated learning. Note that FL is a privacy-preserving framework for training machine learning models. We resort to point process-based models, i.e., Mutual-Exciting Process (MEP) [31], [32], to illustrate how PPVF works. Note that MEP is employed here because it has been widely used in [33], [22], [34] for predicting video utility in traditional online video



caching problems. It is not difficult to replace MEP with a new model for video utility prediction in PPVF. In this section, we focus on how to modify it to fit in PPVF.

#### A. Intensity and Likelihood Function

The core of a point process lies in its intensity function, denoting the occurrence probability of an event within a tiny time window  $[t, t + dt]$  [34]. By abusing notations a little bit, an intensity function can be defined by  $h(\iota, t)dt = P\{\Omega | \mathcal{V}(t)\} = E(dN(\iota, t) | \mathcal{V})$ , where  $N(\iota, t)$  is the count function and  $E(dN(\iota, t) | \mathcal{V})$  represents the expected count of occurrences of event  $\Omega$  with type  $\iota$  in the time window  $[t, t + dt]$  based on the historical event set  $\mathcal{V}$  [34].

In PPVF, the historical event set  $\mathcal{V}$  corresponds to the historical record set of viewing requests. We can create an intensity function for a particular video  $i$  (i.e., event type  $\iota$ ) indicating the expected request rate from users for that video, which is regarded as the utility of video  $i$  for caching. Recall that the set  $\mathcal{T}_{e,i}^{0,t^k}$  represents the timestamps corresponding to requests of video  $i$  in local viewing records  $\mathcal{V}_e^k$  at ED  $e$  within the time interval  $[0, t^k]$ . We can create the intensity function for video  $i$  and time  $t^k$  at ED  $e$  as:

$$h_e(i, t^k | \mathcal{V}_e^k, \beta, \omega) = \beta_i + \sum_{j \in \mathcal{I}} \omega_{i,j} \sum_{\tau \in \mathcal{T}_{e,j}^{0,t^k}} \phi(t^k - \tau), \quad (9)$$

where  $\omega = [\omega_{i,j}]^{I \times I}$ ,  $\omega_{i,j} \in \mathbb{R}^+$  denotes the influencing parameter matrix among all videos, while  $\beta = [\beta_i]^I$ ,  $\beta_i \in \mathbb{R}^+$  is the bias parameter vector of the intensity functions. Specifically,  $\phi(\cdot)$  is defined as  $\phi(t) = \exp(-\delta \cdot t)$ , where exponential decreasing kernel functions are adopted to gauge the influence of historical events for point process models [34], [32], [31]. Here,  $\delta > 0$  serves as a hyper-parameter of the influence attenuation coefficient.

**Remark 2.** The intuition of Eq. (9) is that users' video requests at different time points are correlated, where a more recent historical video request would contribute a higher request rate to its relevant video. The extent can be captured by influencing parameters and kernel functions in the point process model to predict future request rates.

To make our presentation concise,  $h_e(i, t^k)$  is used to represent  $h_e(i, t^k | \mathcal{V}_e^k, \beta, \omega)$  hereafter if the meaning is clear. The parameter space of  $\omega_{i,j}$  in Eq. (9) is  $O(I^2)$  which is prohibitive for solving directly. The parameter space can be reduced by Singular Value Decomposition (SVD) [33]. Given that  $\omega_{i,j}$  represents how much video  $j$  influences video  $i$ , it can be decomposed as the product of  $\omega_{i,j} = \mathbf{p}_i \cdot \mathbf{q}_j^T$ . Here,  $\mathbf{p}_i$  and  $\mathbf{q}_j$  are latent vectors with dimension  $D \ll I$ . Hence, we can significantly shrink the dimension space of  $\omega$  to avoid overfitting. Specifically, the parameter dimensions can be condensed from  $I \times I$  to  $2 \times I \times D$ , where  $D \ll I$ . Consequently, the utility  $\lambda_{e,i}^k, \forall e, i, k$  can be obtained by the revised form in Eq. (10).

$$\lambda_{e,i}^k = \hat{h}_e(i, t^k) = \beta_i + \sum_{j \in \mathcal{I}} \mathbf{p}_i \cdot \mathbf{q}_j^T \sum_{\tau \in \mathcal{T}_{e,j}^{0,t^k}} \phi(t^k - \tau). \quad (10)$$

Next, we can use the maximum likelihood estimation (MLE) [22] to optimize all parameters denoted by  $\theta \stackrel{\text{def}}{=} \{\beta, \mathbf{p}, \mathbf{q}\}$  in Eq. (10). With local timestamp set  $\mathcal{T}_{e,i}$  for video  $i$  in time  $[0, T)$  at ED  $e$ , the local log-likelihood function is derived as:

$$ll_e(\theta | \mathcal{V}_e) = \sum_{i \in \mathcal{I}} \sum_{\tau \in \mathcal{T}_{e,i}} \log \hat{h}_e(i, \tau) - \int_0^T \hat{h}_e(i, t) dt. \quad (11)$$

Here,  $\mathcal{V}_e$  represents the whole private dataset at ED  $e$  to generate the time timestamp  $\mathcal{T}_{e,i}$  for any  $i \in \mathcal{I}$ . The detailed derivation can be found in Appendix B. To preserve privacy, each ED  $e$  should locally maximize Eq. (11). However, the estimation accuracy will be inferior because request records owned by each ED can be very scarce. Moreover, given the dynamic nature of video popularity, parameter estimation cannot be solved by one-time training. Continuous online learning is necessary to closely track the changes in video request patterns. To address these challenges, we propose an FL-based online parameter estimation algorithm, and the CP can coordinate the training process by collecting, aggregating, and distributing model parameters.

#### B. FL-based Online Parameter Estimation

**1) Local Online Log-Likelihood Function for EDs:** In practical video systems, user requests are generated online, which can make the computation complexity of  $\lambda_{e,i}^k$  very heavy. To alleviate computation overhead, we simplify Eq. (10) by removing distant historical events without compromising the accuracy of utility prediction. In Eq. (10), the kernel  $\phi(t - \tau)$  represents the influence of the request at time  $\tau$  on video utility, where  $t$  is the current time. If  $t - \tau \gg 1$ , it implies that  $\phi(t - \tau) \approx 0$  and the influence of the request at the past time  $\tau$  on video utility prediction is negligible. Thus, we set a threshold  $\phi_{th}$  to eliminate these distant records from computation. At time  $t$ , all records before time  $t + \frac{\log \phi_{th}}{\delta}$  will be ignored. In this way, we can significantly reduce computation overhead.

Let  $\Delta t = -\frac{\ln \phi_{th}}{\delta}$ . At time  $t_\theta$ , where  $t_\theta \in \mathcal{Q}$  is the timestamp to update model parameters, we only consider records in the period  $[t_\theta - \Delta t, t_\theta]$  in an online manner.

$$\hat{ll}_e(\theta | \mathcal{V}_e) = \sum_{i \in \mathcal{I}} \left( \sum_{\tau \in \mathcal{T}_{e,i}^{t_\theta - \Delta t, t_\theta}} \log \hat{h}_e(i, \tau) - \int_{t_\theta - \Delta t}^{t_\theta} \hat{h}_e(i, t) dt \right). \quad (12)$$

Notably, unlike the approach in [22], we only truncate the online update interval for the log-likelihood calculation while preserving the influence of all historical events in the intensity function. This choice aligns with our ability to incrementally compute the impact of historical records in Eq. (10), Eq. (12), avoiding excessive computational complexity. For a detailed discussion on the incremental computation of the point process, please consult [33].

With a direct mathematical derivation, the partial derivatives of each parameter respected to Eq. (12) can be derived as follows:

$$\frac{\partial \hat{l}_e(\theta | \mathcal{V}_e)}{\partial \beta_i} = \sum_{\tau \in \mathcal{T}_{e,i}^{t_\theta - \Delta t, t_\theta}} \frac{1}{\hat{h}_e(i, \tau)} - \Delta t, \quad (13)$$

$$\begin{aligned} \frac{\partial \hat{l}_e(\theta | \mathcal{V}_e)}{\partial \mathbf{p}_i} = & \sum_{\tau \in \mathcal{T}_{e,i}^{t_\theta - \Delta t, t_\theta}} \frac{\sum_{j \in \mathcal{I}} \mathbf{q}_j^\top \sum_{\tau' \in \mathcal{T}_{e,j}^{0, \tau}} \phi(\tau - \tau')}{\hat{h}_e(i, \tau)} \\ & - \sum_{j' \in \mathcal{I}} \mathbf{q}_{j'}^\top \left( \sum_{\tau'' \in \mathcal{T}_{e,j'}^{0, t_\theta - \Delta t}} \int_{t_\theta - \Delta t - \tau''}^{t_\theta - \tau''} \phi(t) dt \right. \\ & \left. - \sum_{\tau''' \in \mathcal{T}_{e,j'}^{t_\theta - \Delta t, t_\theta}} \int_0^{t_\theta - \tau'''} \phi(t) dt \right), \end{aligned} \quad (14)$$

$$\begin{aligned} \frac{\partial \hat{l}_e(\theta | \mathcal{V}_e)}{\partial \mathbf{q}_j} = & \sum_{i \in \mathcal{I}} \mathbf{p}_i \left( \sum_{\tau \in \mathcal{T}_{e,i}^{t_\theta - \Delta t, t_\theta}} \frac{\sum_{\tau' \in \mathcal{T}_{e,j}^{0, \tau}} \phi(\tau - \tau')}{\hat{h}_e(i, \tau)} \right. \\ & - \sum_{\tau'' \in \mathcal{T}_{e,j}^{0, t_\theta - \Delta t}} \int_{t_\theta - \Delta t - \tau''}^{t_\theta - \tau''} \phi(t) dt \\ & \left. - \sum_{\tau''' \in \mathcal{T}_{e,j}^{t_\theta - \Delta t, t_\theta}} \int_0^{t_\theta - \tau'''} \phi(t) dt \right). \end{aligned} \quad (15)$$

Upon computing the local likelihood value  $\hat{l}_e(\theta | \mathcal{V}_e)$  and gradient values  $\frac{\partial \hat{l}_e(\theta | \mathcal{V}_e)}{\partial \beta_i}$ ,  $\frac{\partial \hat{l}_e(\theta | \mathcal{V}_e)}{\partial \mathbf{p}_i}$ , and  $\frac{\partial \hat{l}_e(\theta | \mathcal{V}_e)}{\partial \mathbf{q}_j}$  using local historical viewing records from time  $[t_\theta - \Delta t, t_\theta)$ , the EDs transmit only these non-sensitive gradients and likelihood values to the CP for global estimation.

2) **Global Log-Likelihood Function for CP:** For a more precise understanding of the global estimation, we present the global likelihood function as follows:

$$\begin{aligned} \min_{\theta} L = & - \sum_{e \in \mathcal{E}} \hat{l}_e(\theta | \mathcal{V}_e) + \frac{\rho_\beta}{2} \|\beta\|_2^2 + \frac{\rho_p}{2} \|\mathbf{p}\|_2^2 + \frac{\rho_q}{2} \|\mathbf{q}\|_2^2, \\ \text{s.t. } & \beta, \mathbf{q}, \mathbf{p} \in \mathbb{R}^+, \end{aligned} \quad (16)$$

where  $\rho_\beta, \rho_q, \rho_p > 0$  denote regularization parameters. Here, instead of direct computation by the server using private records, all EDs upload the local likelihood function value  $\hat{l}_e(\theta | \mathcal{V}_e)$ . Furthermore, for any  $\theta \in \Theta$ , the CP can aggregate the gradient of  $\theta$  separately, drawing upon the gradient  $\frac{\partial \hat{l}_e(\theta | \mathcal{V}_e)}{\partial \theta}$  provided by the EDs. Subsequently, the partial derivative from Eq. (16), when aggregated in CP, aligns with

$$\frac{\partial L}{\partial \theta} = \rho_\theta \theta + \sum_{e \in \mathcal{E}} \frac{\partial \hat{l}_e(\theta | \mathcal{V}_e)}{\partial \theta}. \quad (17)$$

Let  $\theta^{(n)}$  denote the parameters trained for  $n$  iterations. Subsequently, the update rule for  $\theta$  is:

$$\theta^{(n+1)} \leftarrow \theta^{(n)} + \eta \left( -\frac{\partial L}{\partial \theta^{(n)}} + \rho_\theta \theta^{(n)} \right). \quad (18)$$

Here,  $\rho_\theta$  represents the regularization parameter specific to  $\theta$ , while  $\eta$  signifies the learning rate determined by the selected gradient descent algorithm. Upon completing a round of parameter updates, the updated parameters  $\theta^{(n)}$  are disseminated to each ED for the subsequent iteration.

3) **FL-based Execution:** In our FL framework, EDs compute local likelihood plus gradient values, and subsequently expose these parameters (i.e.,  $\theta$ ) to the parameter server (perhaps maintained by the CP) for parameter aggregation. By interacting with EDs, the CP is responsible for collecting, aggregating, updating, and then disseminating model parameters to EDs. This approach is crafted to optimize the model without sharing raw historical viewing records from EDs, and thus preserves privacy.

It can be completed by iteratively conducting the following two operations on EDs and the CP:

- **EDs' role in FL:** EDs calculate the local likelihood function and gradients using recent timestamp sets  $\cup_{i \in \mathcal{I}} \mathcal{T}_{e,i}^{t_\theta - \Delta t, t_\theta}$  related to historical record set, in conjunction with model parameters  $\theta^{(n)}$  from the  $n$ -th iterations. Then, EDs transmit these gradients and likelihood values to the parameter server. Therefore, user privacy is preserved since the original data will not be exposed.
- **CP's role in FL:** Once the CP receives computations from EDs, it aggregates likelihood functions and gradient values for all EDs. This consolidated result underpins the global gradient update. Following this update, the CP distributes the revised model parameters  $\theta^{(n+1)}$  to EDs.

The details can be found in Alg. 3 in Appendix C.

## V. PERFORMANCE EVALUATION

In this section, we conduct trace-driven experiments to evaluate PPVF using a real viewing dataset in Tencent Video. We seek to answer the following three questions: (1) How well do PPVF's components of *budget scheduler* and *request generator* perform in distorting the private information in user profiles (Section V-D1)? (2) How well does PPVF work to protect users' interests against the powerful recommendation system (Section V-D2)? (3) How well can the PPVF framework adapt to traffic changes and improve edge caching performance compared to fixed experts and SOTA learning-based approaches (Section V-E)?

To facilitate the peer review, we also release the source code of our system PPVF<sup>2</sup> and the dataset<sup>3</sup>. We now discuss the methodology and setup of our evaluations.

### A. Dataset and Settings

Given the large scale of Tencent Video Datasets, we randomly sample a small subset of 10,000 users drawn from the upper echelon of active users (i.e., the set of 20% users with the largest interactive viewing records) in the origin public dataset [22]. The new dataset consists of 933,541 video viewing requests for 10,373 unique videos, all within a specific city over 30 days. Following [33], we evenly group

<sup>2</sup><https://github.com/zhangxzh9/PPVF-MAINCODE>

<sup>3</sup><https://github.com/zhangxzh9/PPVF-DATASET>



these users into 25 fixed groups to replicate a real online video system aided by 25 EDs during the whole 30 days. It is important to note that this number set is simulation-based, and this setting can be tuned by the customized strategy that is adaptable to various scenarios and meets different user privacy requirements. The experimental results also demonstrate the robustness of our framework at different levels of EDs. As such, each request record in the dataset is collated by the metadata  $(e, u, i, \tau)$ .

Similar to [22], the time interval in our experiments is quantized at 1 hour for all requests, i.e., the requests arrived at the same hour have the same timestamp  $\tau$ . It is important to emphasize that the pre-fetching video's timing is not tied to a specific time slot. If a request is not met at the edge, EDs must promptly retrieve the video from CP using the pre-fetching algorithm. Additionally, we split the dataset into two date-based subsets. The initial subset, containing requests with time  $0 \leq \tau < 240$  hours, is used to initialize the system. The subsequent subset with requests over the next 20 days (i.e.,  $240 \leq \tau < 720$  hours) functions as the test period.

Other experimental setups are described according to different tasks: (1) **Point process and FL**: Following [33], the decay parameter and the dimension of the latent vector are designated as  $\delta = 0.01$  and  $D = 10$ , respectively. In alignment with [22], all model parameters (i.e.,  $\beta$ ,  $p$ , and  $q$ ) are initialized at 1.0. For online FL-based parameter estimation, the maximum iteration count is 20 and the truncated threshold is set to  $\phi_{th} = e^{-0.48}$ . Therefore, the interval  $t_\theta$  to update  $\beta$ ,  $p$ , and  $q$  is 2 days (48 hours), which is the same as that setting in [22]. Some detailed experiments are conducted to study the influence of the online updated interval of the FL framework. (2) **Edge pre-fetching and caching**: For experimental consistency, the privacy cost  $(\epsilon_{e,i}, \forall e, i)$  is uniformly set to 1 for each video's pre-fetching requests [35]. Additionally, we standardize the allocation of the privacy budget, pre-fetching capacity, and caching capacity across all EDs with values set at  $\xi_e = 15$ ,  $f_e = 4$ , and  $c_e = 1\%$ , respectively. In specific experiments, one of these parameters might be varied to assess its impact, with the other two being constant.

## B. Baselines

We compare PPVF with three types of baselines. The first type includes privacy-preserving video fetching algorithms, while the second and third types are video caching algorithms that do not consider privacy leakage. Privacy-preserving caching algorithms include: (1) **SAGE** [36], which pre-fetches videos with randomly assigned privacy budgets until the privacy budgets reach the maximum constraints; (2) **BESTFIT**, which allocates the privacy budget to pre-fetch videos with the highest utility until the privacy budgets reach the maximum constraints. Note that these two baselines are only designed for privacy allocation without designing a video utility predictor. For a fair comparison, we implement our video utility predictor in SAGE and BESTFIT for caching.

To further demonstrate the superiority of our utility predictor, we replicate two advanced caching utility prediction methods at the edge for comparison. These algorithms are

introduced as follows: (3) **MAV** [37], which caches the videos at the edge nodes considering the strength of user requests in the future round within the dynamic Stackberg game. The caching utility is calculated by the moving average value (MAV) method, and the weight of MAV is set as 0.9 based on [37]; (4) **HRS** [22], which serves as a video popularity prediction model designed for the edge server, employing a fusion of three distinct point process models. All parameter configurations within this baseline align with the defaults specified in [22]. It is worth mentioning that these two baselines are mainly designed to improve edge caching efficiency with utility (e.g., popularity) prediction. Both of them overlook the privacy of users exposed by pre-fetching requests. Therefore, we only replace our utility predictor module with MAV and HRS and keep the other system components unchanged.

We also compare PPVF with the following two eviction caching algorithms, in which EDs only fetch videos watched by users when the cache is missed. These algorithms include: (5) **LRU (Least Recently Used)**, which replaces the video that has not received any request for the longest time with a newly requested video; (6) **LFU (Least Frequently Used)**, which replaces the video that has been requested in the least number of times with a newly requested video. These two conventional caching algorithms are extensively employed both in industry and academia, making them suitable benchmarks for comparing caching performance.

## C. Metrics

To evaluate PPVF, we employ three metrics to evaluate both privacy protection and system efficiency. More specifically, we adopt the following metrics in our experiments:

- 1) **JS (Jaccard Similarity)** measures the averaged similarity between users' real profiles and profiles exposed by their ED for video fetching. Each profile is represented by a vector of dimension  $I$ , where each element indicates whether a video has been requested by a user or ED during the entire testing period. A lower similarity is more desirable, implying stronger privacy protection.
- 2) **RHR (Recommendation Hit Rate) Degradation**, which calculates the averaged degradation of RHR among all users when using a recommendation algorithm to recommend videos for users based on their original profiles and noisy profiles exposed by EDs. A larger degradation of RHR implies stronger privacy protection. A popular collaborative filtering recommendation algorithm [26], NCF, is implemented with the same settings in [26] as the adversary in our experiments.
- 3) **CHR (Cache Hit Ratio)**, which is defined as the number of video hits at all EDs divided by the total number of original video requests from users over the entire test period. CHR is employed to evaluate the caching system efficiency.

## D. Effectiveness of Privacy Protection

We first evaluate the performance of privacy protection using two metrics, i.e., the average JS and the degradation of RHR. We then further investigate the final status of the remaining privacy budget of all content.

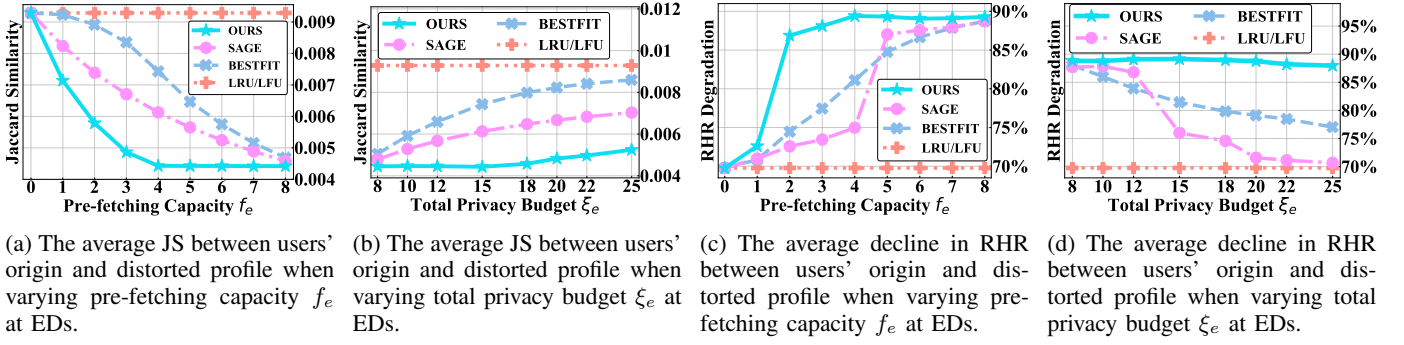


Fig. 2: Comparing privacy protection performances under different metrics for PPVF and other baselines with different settings of pre-fetching capacity or total privacy budget. A lower PDR and a larger RHR degradation are preferred by users to protect their private viewing profiles.

1) **Effectiveness of Distorting Private Information:** For experiments presented in Fig. 2a and Fig. 2b, we compare the average JS between users' original profiles and profiles exposed by EDs after the test period. In Fig. 2a, we fix  $c_e = 1\%$  and  $\xi_e = 15$ , but vary  $f_e$  to study the average JS under different numbers of redundant requests. Whereas, in Fig. 2b, we vary  $\xi_e$  but fix  $c_e = 1\%$  and  $f_e = 4$  to study privacy protection under different privacy budget of EDs. As presented in Fig. 2a and Fig. 2b, we can observe that PPVF steadily outperforms other baselines. It exhibits an average reduction of 17.54% (22.38%) of JS compared to the second-best privacy-preserving baseline when varying the pre-fetching capacity (or the total privacy budget) for each ED.

These experimental results manifest that PPVF can significantly distort exposed user profiles so that users' video request privacy can be preserved. Compared with SAGE and BESTFIT, PPVF achieves the lowest average JS because PPVF considers both limited privacy budget and video utility when pre-fetching videos. Recall that we tune a threshold to select videos for generating redundant requests in Alg. 1. When the privacy budget is plentiful, PPVF selects videos of high utility with higher priority. However, if the privacy budget of the video is tight, we tune the threshold so that PPVF can select the video more conservatively. Instead, more diversified videos will be selected to conceal user privacy. Note that the average JS of classical caching algorithms, i.e., LRU, and LFU, are also compared with ours. Although these algorithms do not consider privacy protection with the redundant fetching videos, they can benchmark the degree of protection offered by privacy-preserving algorithms at edge devices.

2) **Privacy Protection against Recommendation Systems:** We further employ the degradation of RHR to evaluate privacy protection by implementing the algorithm in [26] to recommend videos based on request records exposed by EDs after the test period. The configurations in Fig. 2c and Fig. 2d mirror those in Fig. 2a and Fig. 2b, respectively. By using original user profiles for a recommendation, the algorithm in [26] can achieve 99.42% RHR, indicating the effectiveness of the recommendation. Then, the Degradation of RHR calculates the gap between the accuracy achieved by utilizing request records exposed by EDs and the original accuracy 99.42%. The experimental results in Fig. 2c and Fig. 2d indicate that:

- PPVF is the best one to achieve the highest RHR degra-

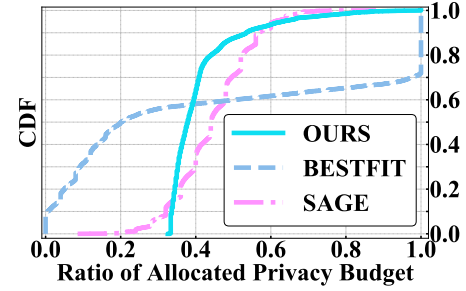


Fig. 3: The CDF (cumulative distribution function) of the average ratio of the allocated privacy budget for videos at the end of the test.

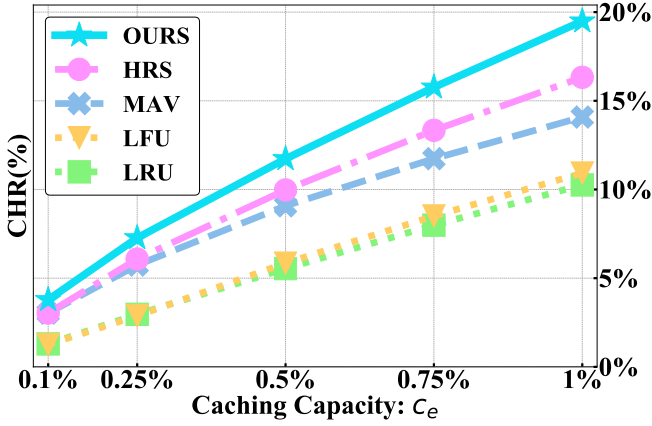
dation under all different scenarios. In particular, the performance of PPVF is better when the pre-fetching capacity is limited or the total privacy budget is sufficient.

- SAGE and BESTFIT are better than LRU/LFU in privacy-preservation. However, their performance is inferior to PPVF under the same constraints, e.g., pre-fetching capacity and privacy budget.
- LRU/LFU can degrade RHR performance because they are implemented on EDs, which only expose consolidated request records of multiple users, making it difficult for recommenders to identify personalized interests.

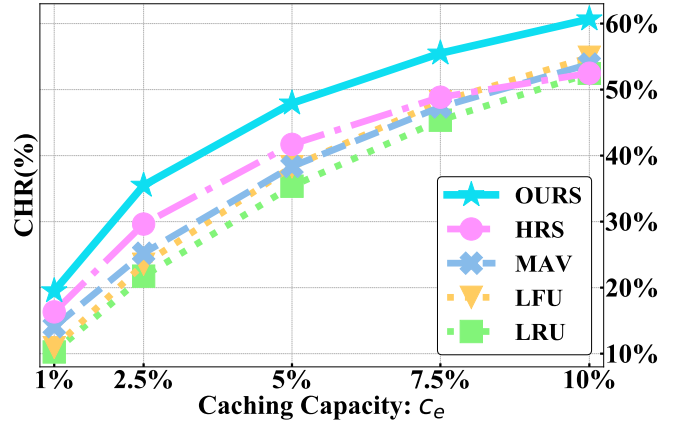
3) **Remaining Privacy Budgets of all Content:** In Fig. 3, we plot the cumulative distribution function (CDF) of the remaining privacy budgets of all videos after the test period to visualize redundant request decisions made by different caching algorithms. Here, we set  $\xi_e = 15$  by default. From Fig. 3, we can observe that PPVF can use up privacy budgets of videos worth for caching, and thereby, there are nearly 70% videos with 60% residual privacy budget at the end of the test. In contrast, the budget consumption of BESTFIT and SAGE is scattered among different videos. Their redundant requests for the hottest or coldest videos are not effective in preserving privacy, which is why their protection is weaker than ours.

### E. Effective Caching Performance

To comprehensively demonstrate the superiority of PPVF, we also compare CHR performance between different algorithms. In this experiment, we tune the caching capacity of



(a) The average CHR for all EDs in some small caching capacities.



(b) The average CHR for all EDs in some large caching capacities.

Fig. 4: Comparing caching performance for PPVF and other baselines when varying the caching capacity  $c_e$  of all EDs. For more detailed results, please refer to Table III in Appendix D.

each ED from 0.1% to 10% of the total video number, i.e., the caching capacity ranges from 10 to 1037 videos. For simplicity, we ignore the size difference of videos [22]. We calculate the CHR of the entire system over the test period. The results are plotted in Fig. 4, in which the y-axis represents the average CHR and the x-axis represents the caching capacity of each ED. To better show the difference between PPVF, SAGE, and BESTFIT, we present numerical results in Table III in Appendix D. From the experimental results in Fig. 4 and Table III, we can observe that:

- PPVF is slightly better than SAGE and BESTFIT in terms of the CHR performance among the small capacities, while BESTFIT achieves the highest CHR when the caching capacity is large. Note that it is fair to compare the CHR performance between PPVF, SAGE, and BESTFIT because all algorithms cache redundant video requests based on the same predicted utility.
- Except for SAGE and BESTFIT, our PPVF consistently attains the highest CHR in comparison to other caching baselines. This translates to an average enhancement of 18.15% over the second-best caching algorithm, HRS, within all capacity settings. The presented results demonstrate the robustness of PPVF, indicating its potential applicability for implementation across heterogeneous edge devices with varying caching capacities.
- On the one hand, PPVF outperforms HRS / MAV in CHR by leveraging a more effective utility prediction method to reliably aggregate the private information at all EDs based on the FL framework. On the other hand, the CHR performance of PPVF surpasses that of LFU/LRU because these eviction-based algorithms do not make any redundant video requests to improve their caching efficiency. The superior performance of PPVF over these SOTA baselines can be attributed to its ability to request and cache high-utility redundant videos, thereby elevating the CHR.
- Moreover, PPVF demonstrates a more efficient utilization of caching capacity. To elucidate, when working with a limited caching capacity of 0.1%, PPVF's performance

TABLE II: The average CHR (%) results under different  $t_\theta$  (hours). All Settings are on default as described in Sec. V-A except varying except  $t_\theta$ .

$c_e$	$t_\theta$ (hours)					
	12	24	48	72	120	240
0.1%	3.944	<b>3.975</b>	3.782	3.471	3.251	2.472
1%	19.01	19.40	<b>19.49</b>	19.37	18.74	17.48
10%	58.07	60.37	<b>60.68</b>	60.36	60.53	59.06

surpasses the second-best caching solution by more than 24.70%. This is especially notable under constrained caching resources, underscoring PPVF's exceptional capability to predict the most popular videos, even if using distorted information in the online FL framework. With more accurately estimated video utility, PPVF can accordingly pre-fetch videos to attain superior CHR performance.

Lastly, we study the sensitivity of the online parameter update interval  $t_\theta$  in Table II to see how this hyper-parameter affects the video caching performance. All other hyper-parameters are kept unchanged as we vary  $t_\theta$ . As illustrated in Table II, the comprehensive CHR exhibits an effective improvement when this hyper-parameter is minimized. A smaller  $t_\theta$  means a more frequent online parameter update in the FL framework. This observation stems from the fact that a more frequent update contributes to sustaining an efficient model for predicting video utility. Nevertheless, this enhancement comes at the expense of an increased computational burden. Considering this trade-off, the selection of a 2-day (48-hour) interval for parameter updates in previous work [22] and our study is deemed rational.

## VI. RELATED WORK

User privacy, including historical records, location, and other personal information, is an important concern in online video systems, prompting significant research efforts for

safeguarding it. Various approaches have been introduced to address this concern. For instance, noise-based methods like DP [38] and Anonymous [2], [6] have been proposed to shield location information, while blockchain-based techniques [39], [40] have been employed to safeguard users' personal information. Despite these efforts, the focus on request privacy, deemed the most critical user privacy aspect [4], is also essential during the design of privacy-preserving video systems. In this section, we briefly review existing relevant works from two perspectives: privacy leakage in online video services and its protection.

#### A. Request Privacy Leakage

Privacy concerns in online video services span multiple dimensions: request traces [11], [10], personal details [41], [8], location data [6], [3], [2], and specific content data [17], [7], to name a few. Among these, request traces have emerged as particularly pivotal within online video services, as they may inadvertently expose user preferences to potential adversaries [4]. Such traces frequently encapsulate sensitive user information, capturing browsing patterns, preferences, and interests of users [10], [7]. Commercial motivations propel content providers to amass and scrutinize users' private data [16]. This collected data is multifaceted, comprising geographical locations [3], behavioral tendencies [42], personal specifics [12], among other aspects. Leveraging this data can substantially refine service quality for content providers across domains, including content caching [33], recommendation engines [23], and video distribution [43], [44].

#### B. Request Privacy Protection at the Network Edge

Privacy protection in online video services at the network edge can be broadly classified into three categories. The *first* focuses on cryptography based techniques, including encryption transmission [17], [45] and blockchain-enabled methods [5], [46]. While these techniques are potent, they impose significant computational demands on edge devices and cannot entirely prevent CPs from potential misuse of user data. The *second* category encompasses trusted distributed computing (TDC) techniques, exemplified by federated learning [9], [20], [47]. Although these methods bolster user privacy by obviating the need for direct data transfer, their suitability for online video platforms is debatable, given their limited capability to prohibit content providers from tracking user viewing habits. The *third* category is grounded in noise-based techniques. These methods accentuate request privacy within edge networks by obfuscating actual user preferences [24], [48]. A prevalent approach within this category is the pre-fetching of redundant and unrelated videos to foster ambiguity. Such pre-fetched content can also be cached at the network edge to serve future requests, thereby curtailing direct interactions with CPs [49], [50], which in turn mitigates the data exposure risk. Nevertheless, balancing the quality of edge services with the imperative of user data protection remains an intricate endeavor.

one viable solution to ensure request privacy in online video systems involves incorporating DP noises, which delivers robust information protection assurances [13], [4], [12]. L'ecuyer

et al. [36] pioneered the use of block composition to tackle privacy concerns arising from expanding private datasets. This innovative method provides theoretical assurances for the efficient utilization of individual dataset segments. Moreover, to shield non-iid datasets, correlated differential privacy was introduced in [51], [27], taking into account the interdependence among records. Yet, the challenge confronted by allocating limited privacy budgets for online video requests on edge devices persists. Certain allocation frameworks, like Sage [36] and DP-FLames [52], may be overly simplistic or rely on improbable assumptions, thereby restricting their flexibility in diverse scenarios. In light of these methodological limitations, we present a novel privacy protection strategy. This method enhances request privacy by generating redundant requests, all while preserving the operational efficacy of edge caching.

## VII. CONCLUSION

With the proliferation of online video services, preserving request privacy remains an open problem. The challenge of this problem lies in that online video providers can automatically capture video requests from users. As a consequence, user requests cannot be trivially distorted by injecting noises or protected with encryption. In this work, we are among the first to attempt to address this challenge by proposing the PPVF framework, which synthetically utilizes trusted edge caching, correlated differential privacy, and federated learning. In other words, edge devices try to conceal user request privacy by generating noisy requests (with the noise scale calibrated according to video correlation) to the video provider. To maintain system efficiency, edge devices collaboratively predict video utility via FL so that they can harmonize video utility and privacy leakage amount when requesting videos. With the advancement of the online video market, privacy-preserving techniques presented in this work offer invaluable insights and solutions for bolstering user privacy when consuming video content. Subsequent endeavors can build upon these foundations to further propel the field of privacy-preserving online video services.

## REFERENCES

- [1] Global Media Insight, "Youtube statistics 2024 (demographics, users by country & more )," p. 1, 2024. [Online]. Available: <https://www.globalmediainsight.com/blog/youtube-users-statistics/>
- [2] N. Nisha, I. Natgunanathan, S. Gao, and Y. Xiang, "A novel privacy protection scheme for location-based services using collaborative caching," *Computer Networks*, vol. 213, p. 109107, Aug 2022.
- [3] S. Amini, J. Lindqvist, J. Hong, J. Lin, E. Toch, and N. Sadeh, "Cache: Caching location-enhanced content to improve user privacy," in *Proceedings of the 9th International Conference on Mobile Systems, Applications, and Services (MobiCom)*, no. 11. ACM, 2011, pp. 197–209.
- [4] M. Wang, C. Xu, X. Chen, H. Hao, L. Zhong, and S. Yu, "Differential privacy oriented distributed online learning for mobile social video prefetching," *IEEE Transactions on Multimedia*, vol. 21, no. 3, pp. 636–651, Jan 2019.
- [5] Y. Qian, Y. Jiang, L. Hu, M. S. Hossain, M. Alrashoud, and M. Al-Hammadi, "Blockchain-based privacy-aware content caching in cognitive internet of vehicles," *IEEE Network*, vol. 34, no. 2, pp. 46–51, 2020.
- [6] L. Hu, Y. Qian, M. Chen, M. S. Hossain, and G. Muhammad, "Proactive Cache-Based Location Privacy Preserving for Vehicle Networks," *IEEE Wireless Communications*, vol. 25, no. 6, pp. 77–83, Dec 2018.

- [7] J. Cui, L. Wei, H. Zhong, J. Zhang, Y. Xu, and L. Liu, "Edge computing in VANETs-An efficient and privacy-preserving cooperative downloading scheme," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 6, pp. 1191–1204, Apr 2020.
- [8] X. Zhang, H. Zhong, C. Fan, I. Bolodurina, and J. Cui, "CBACS: A Privacy-Preserving and Efficient Cache-Based Access Control Scheme for Software Defined Vehicular Networks," *IEEE Transactions on Information Forensics and Security*, vol. 17, pp. 1930–1945, May 2022.
- [9] D. Qiao, S. Guo, D. Liu, S. Long, P. Zhou, and Z. Li, "Adaptive Federated Deep Reinforcement Learning for Proactive Content Caching in Edge Computing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 33, no. 12, pp. 4767–4782, Dec 2022.
- [10] V. Sivaraman and B. Sikdar, "A Defense Mechanism against Timing Attacks on User Privacy in ICN," *IEEE/ACM Transactions on Networking*, vol. 29, no. 6, pp. 2709–2722, Dec 2021.
- [11] W. Tong, W. Chen, B. Jiang, F. Xu, Q. Li, and S. Zhong, "Privacy-Preserving Data Integrity Verification for Secure Mobile Edge Storage," *IEEE Transactions on Mobile Computing*, vol. Early Acce, pp. 1–1, Mar 2022.
- [12] P. Zhou, K. Wang, J. Xu, and D. Wu, "Differentially-private and trustworthy online social multimedia big data retrieval in edge computing," *IEEE Transactions on Multimedia*, vol. 21, no. 3, pp. 539–554, Mar 2019.
- [13] Q. Cai, Z. Xue, C. Zhang, W. Xue, S. Liu, R. Zhan, X. Wang, T. Zuo, W. Xie, D. Zheng, P. Jiang, and K. Gai, "Two-Stage Constrained Actor-Critic for Short Video Recommendation," in *Proceedings of the ACM Web Conference 2023 (WWW '23)*. ACM, Apr 2023, pp. 865–875.
- [14] Q. Yang and P. Kong, "RuleCache: A mobility pattern based multi-level cache approach for location privacy protection," in *2016 IEEE 22nd International Conference on Parallel and Distributed Systems (ICPADS)*. IEEE, 2016, pp. 448–455.
- [15] L. Xiao, X. Wan, C. Dai, X. Du, X. Chen, and M. Guizani, "Security in Mobile Edge Caching with Reinforcement Learning," *IEEE Wireless Communications*, vol. 25, no. 3, pp. 116–122, Jun 2018.
- [16] J. Ni, K. Zhang, and A. V. Vasilakos, "Security and Privacy for Mobile Edge Caching: Challenges and Solutions," *IEEE Wireless Communications*, vol. 28, no. 3, pp. 77–83, Jun 2021.
- [17] A. Araldo, G. Dan, and D. Rossi, "Caching Encrypted Content Via Stochastic Cache Partitioning," *IEEE/ACM Transactions on Networking*, vol. 26, no. 1, pp. 548–561, Jan 2018.
- [18] J. Liu, J. Liu, H. Xu, Y. Liao, Z. Wang, and Q. Ma, "Yoga: Adaptive layer-wise model aggregation for decentralized federated learning," *IEEE/ACM Transactions on Networking*, vol. 32, no. 2, pp. 1768–1780, 2024.
- [19] T. Nguyen and M. T. Thai, "Preserving privacy and security in federated learning," *IEEE/ACM Transactions on Networking*, vol. 32, no. 1, pp. 833–843, 2024.
- [20] X. Liu, Z. Yan, Y. Zhou, D. Wu, X. Chen, and J. H. Wang, "Optimizing parameter mixing under constrained communications in parallel federated learning," *IEEE/ACM Transactions on Networking*, vol. 31, no. 6, pp. 2640–2652, 2023.
- [21] J. Hu, Z. Wang, Y. Shen, B. Lin, P. Sun, X. Pang, J. Liu, and K. Ren, "Shield against gradient leakage attacks: Adaptive privacy-preserving federated learning," *IEEE/ACM Transactions on Networking*, vol. 32, no. 2, pp. 1407–1422, 2024.
- [22] X. Zhang, Y. Zhou, D. Wu, M. Hu, X. Zheng, M. Chen, and S. Guo, "Optimizing Video Caching at the Edge: A Hybrid Multi-Point Process Approach," *IEEE Transactions on Parallel and Distributed Systems*, vol. 33, no. 10, pp. 2597–2611, Oct 2022.
- [23] R. Guerraoui, A. M. Kermarrec, and M. Taziki, "The utility and privacy effects of a click," in *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'17)*. ACM, Aug 2017, pp. 665–674.
- [24] A. A. Sen, F. B. Eassa, M. Yamin, and K. Jambri, "Double Cache Approach with Wireless Technology for Preserving User Privacy," *Wireless Communications and Mobile Computing*, vol. 2018, pp. 1–11, 2018.
- [25] Y. Zhou, L. Chen, C. Yang, and D. M. Chiu, "Video Popularity Dynamics and Its Implication for Replication," *IEEE Transactions on Multimedia*, vol. 17, no. 8, pp. 1273–1285, Aug 2015.
- [26] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T. S. Chua, "Neural collaborative filtering," in *Proceedings of the 26th International Conference on World Wide Web (WWW'17)*. ACM, 2017, pp. 173–182.
- [27] T. Zhu, P. Xiong, G. Li, and W. Zhou, "Correlated differential privacy: Hiding information in Non-IID data set," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 2, pp. 229–242, Feb 2015.
- [28] W. Li, L. Xiang, B. Guo, Z. Li, and X. Wang, "DPlanner: A Privacy Budgeting System for Utility," *IEEE Transactions on Information Forensics and Security*, vol. 18, pp. 1196–1210, Dec 2022.
- [29] W. Li, L. Xiang, Z. Zhou, and F. Peng, "Privacy budgeting for growing machine learning datasets," in *Proceedings - IEEE INFOCOM 2021*. IEEE, May 2021, pp. 1–10.
- [30] F. McSherry, "Privacy integrated queries: An extensible platform for privacy-preserving data analysis," in *Proceedings of the International Conference on Management of Data and 28th Symposium on Principles of Database Systems (SIGMOD-PODS'09)*. ACM, 2009, pp. 19–30.
- [31] D. Daley, Daryl J and Vere-Jones, *An introduction to the theory of point processes: volume II: general theory and structure*, ser. Probability and Its Applications. Springer, 2008, vol. 6(13).
- [32] A. G. Hawkes, "Spectra of Some Self-Exciting and Mutually Exciting Point Processes," *Biometrika*, vol. 58, no. 1, p. 83, Apr 1971.
- [33] Z. Shi, Y. Zhou, D. Wu, and C. Wang, "PPVC: Online Learning Toward Optimized Video Content Caching," *IEEE/ACM Transactions on Networking*, vol. 30, no. 3, pp. 1029–1044, Jun 2022.
- [34] M.-A. Rizoio, Y. Lee, S. Mishra, and L. Xie, *Hawkes processes for events in social media*. Association for Computing Machinery and Morgan & Claypool, 2017, p. 191–218. [Online]. Available: <https://doi.org/10.1145/3122865.3122874>
- [35] K. Pan and K. Feng, "Differential privacy-enabled multi-party learning with dynamic privacy budget allocating strategy," *Electronics*, vol. 12, no. 3, 2023.
- [36] M. Lécluyer, R. Spahn, K. Vodrahalli, R. Geambasu, and D. Hsu, "Privacy accounting and quality control in the sage differentially private ML platform," in *Proceedings of the 27th ACM Symposium on Operating Systems Principles (SOSP'19)*. ACM, Oct 2019, pp. 181–195.
- [37] X. Zhang, L. Xiao, Y. Zhou, M. Hu, D. Wu, and G. Liu, "crrv: A stackelberg game approach for joint privacy-aware video requesting and edge caching," *arXiv preprint arXiv:2310.12622*, 2023. [Online]. Available: <http://arxiv.org/abs/2310.12622>
- [38] Z. Zhang, T. Cao, X. Wang, H. Xiao, and J. Guan, "VC-PPQ: Privacy-preserving Q-learning Based Video Caching Optimization in Mobile Edge Networks," *IEEE Transactions on Network Science and Engineering*, vol. 9, no. 6, pp. 4129–4144, Aug 2022.
- [39] Y. Dai, D. Xu, K. Zhang, S. Maharjan, and Y. Zhang, "Deep Reinforcement Learning and Permissioned Blockchain for Content Caching in Vehicular Edge Computing and Networks," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 4, pp. 4312–4324, Apr 2020.
- [40] L. Cui, X. Su, Z. Ming, Z. Chen, S. Yang, Y. Zhou, and W. Xiao, "CREAT: Blockchain-Assisted Compression Algorithm of Federated Learning for Content Caching in Edge Computing," *IEEE Internet of Things Journal*, vol. 9, no. 16, pp. 14 151–14 161, Aug 2022.
- [41] K. Xue, P. He, X. Zhang, Q. Xia, D. S. Wei, H. Yue, and F. Wu, "A Secure, Efficient, and Accountable Edge-Based Access Control Framework for Information Centric Networks," *IEEE/ACM Transactions on Networking*, vol. 27, no. 3, pp. 1220–1233, Jun 2019.
- [42] Y. Zhang, P. Zhao, K. Bian, Y. Liu, L. Song, and X. Li, "DRL360: 360-degree Video Streaming with Deep Reinforcement Learning," in *Proceedings - IEEE INFOCOM 2019*. IEEE, Apr 2019, pp. 1252–1260.
- [43] H. Gupta, J. Chen, B. Li, and R. Srikant, "Online Learning-Based Rate Selection for Wireless Interactive Panoramic Scene Delivery," in *Proceedings - IEEE INFOCOM 2022*. IEEE, Jun 2022, pp. 1799–1808.
- [44] V. Kirilin, A. Sundarajan, S. Gorinsky, and R. K. Sitaraman, "RL-Cache: Learning-Based Cache Admission for Content Delivery," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 10, pp. 2372–2385, Oct 2020.
- [45] Q. Xu, Z. Su, Q. Zheng, M. Luo, B. Dong, and K. Zhang, "Game theoretical secure caching scheme in multihoming edge computing-enabled heterogeneous networks," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4536–4546, Jun 2019.
- [46] Y. Jiang, Y. Zhong, and X. Ge, "IIoT Data Sharing Based on Blockchain: A Multi-leader Multifollower Stackelberg Game Approach," *IEEE Internet of Things Journal*, vol. 9, no. 6, pp. 4396–4410, Mar 2022.
- [47] X. Liu, Z. Zhong, Y. Zhou, D. Wu, X. Chen, M. Chen, and Q. Z. Sheng, "Accelerating federated learning via parallel servers: A theoretically guaranteed approach," *IEEE/ACM Transactions on Networking*, vol. 30, no. 5, pp. 2201–2215, 2022.
- [48] K. Wang and N. Deng, "A Privacy-Protected Popularity Prediction Scheme for Content Caching Based on Federated Learning," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 9, pp. 10 191–10 196, Jun 2022.
- [49] Q. Wu, Z. Li, G. Tyson, S. Uhlig, M. A. Kaafar, and G. Xie, "Privacy-Aware Multipath Video Caching for Content-Centric Networks," *IEEE*



*Journal on Selected Areas in Communications*, vol. 34, no. 8, pp. 2219–2230, Aug 2016.

- [50] S. Nikolaou, R. Van Renesse, and N. Schiper, “Proactive Cache Placement on Cooperative Client Caches for Online Social Networks,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 4, pp. 1174–1186, Apr 2016.
- [51] J. Chen, H. Ma, D. Zhao, and L. Liu, “Correlated Differential Privacy Protection for Mobile Crowdsensing,” *IEEE Transactions on Big Data*, vol. 7, no. 4, pp. 784–795, Dec 2017.
- [52] Y. Hu, W. Liang, R. Wu, K. Xiao, W. Wang, X. Li, J. Liu, and Z. Qin, “Quantifying and Defending against Privacy Threats on Federated Knowledge Graph Embedding,” in *Proceedings of the ACM Web Conference 2023 (WWW '23)*. ACM, 2023, pp. 2306–2317.



**Xianzhi Zhang** received his B.S. degree from Nanchang University (NCU), Nanchang, China, in 2019 and an M.S. degree from the School of Computer Science and Engineering, Sun Yat-sen University (SYSU), Guangzhou, China, in 2022. He is currently pursuing a Ph.D. degree in the School of Computer Science and Engineering at Sun Yat-sen University, Guangzhou, China. He is also working as a visiting PhD student at the School of Computing, Macquarie University, Sydney, Australia. Xianzhi's current research interests include video caching, caching privacy,

machine learning, and edge computing. His research has been published at IEEE TPDS and won the Best Paper Award at PDCAT 2021.



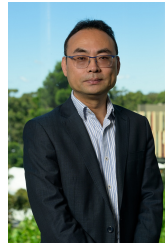
**Yipeng Zhou** is a senior lecturer in computer science at the School of Computing, Macquarie University, and the recipient of ARC (Australian Research Council) DECRA in 2018. From Aug. 2016 to Feb. 2018, he was a research fellow with the Institute for Telecommunications Research (ITR) of the University of South Australia. From 2013.9 to 2016.9, He was a lecturer at the College of Computer Science and Software Engineering at Shenzhen University. He was a Postdoctoral Fellow with the Institute of Network Coding (INC) of the Chinese University of

Hong Kong (CUHK) from Aug. 2012 to Aug. 2013. He obtained his PhD degree and MPhil degree from the Information Engineering (IE) Department of CUHK, respectively. He got a Bachelor's degree in Computer Science from the University of Science and Technology of China (USTC). His research interests include federated learning, privacy protection, and caching algorithm design in networks. He has published more than 80 papers including IEEE INFOCOM, ICNP, IWQoS, IEEE ToN, JSAC, TPDS, TMC, TMM, etc.



**Di Wu** (M'06-SM'17) received his B.S. degree from the University of Science and Technology of China, Hefei, China, in 2000, the M.S. degree from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, in 2003, and the Ph.D. degree in computer science and engineering from the Chinese University of Hong Kong, Hong Kong, in 2007. He was a Post-Doctoral Researcher with the Department of Computer Science and Engineering, Polytechnic Institute of New York University, Brooklyn, NY, USA, from 2007 to 2009, advised

by Prof. K. W. Ross. Dr. Wu is currently a Professor and the Associate Dean of the School of Computer Science and Engineering at Sun Yat-sen University, Guangzhou, China. He was the recipient of the IEEE INFOCOM 2009 Best Paper Award and IEEE Jack Neubauer Memorial Award. His research interests include edge/cloud computing, multimedia communication, Internet measurement, and network security.



**Quan Z. Sheng** is a Distinguished Professor and Head of the School of Computing at Macquarie University, Sydney, Australia. Before moving to Macquarie University, Michael spent 10 years at School of Computer Science, the University of Adelaide, serving in senior leadership roles such as Interim Head of School and Deputy Head of School. Michael holds a PhD degree in computer science from the University of New South Wales (UNSW) and did his post-doc as a research scientist at CSIRO ICT Centre. From 1999 to 2001, Michael also worked at

UNSW as a visiting research fellow. Prior to that, he spent 6 years as a senior software engineer in industries.

Prof. Michael Sheng's research interests include Web of Things, Internet of Things, Big Data Analytics, Web Science, Service-oriented Computing, Pervasive Computing, and Sensor Networks. He is ranked by Microsoft Academic as one of the Most Impactful Authors in Services Computing (ranked Top 5 all time worldwide) and in Web of Things (ranked Top 20 all time). He is the recipient of the AMiner Most Influential Scholar Award on IoT (2019), ARC (Australian Research Council) Future Fellowship (2014), Chris Wallace Award for Outstanding Research Contribution (2012), and Microsoft Research Fellowship (2003). Prof Michael Sheng is Vice Chair of the Executive Committee of the IEEE Technical Community on Services Computing (IEEE TCSVC), the Associate Director (Smart Technologies) of Macquarie's Smart Green Cities Research Centre, and a member of the ACS (Australian Computer Society) Technical Advisory Board on IoT.



**Miao Hu** (S'13-M'17) is currently an Associate Professor at the School of Computer Science and Engineering at Sun Yat-Sen University, Guangzhou, China. He received a B.S. degree and a Ph.D. degree in communication engineering from Beijing Jiaotong University, Beijing, China, in 2011 and 2017, respectively. From September 2014 to September 2015, he was a Visiting Scholar at the Pennsylvania State University, PA, USA. His research interests include edge/cloud computing, multimedia communication, and software-defined networks.



**Linchang Xiao** received his B.S. Degree from the School of Computer Science and Engineering, Sun Yat-sen University (SYSU), Guangzhou, China, in 2022. He is currently working toward his M.S. degree at the School of Computer Science and Engineering, Sun Yat-sen University (SYSU), Guangzhou, China. His research interests include cloud & edge computing, content caching, and multimedia communication.

## APPENDIX A PROOF OF THEOREM 1

Before the proof, the detailed algorithm for online video selection and privacy budget allocation is shown in Alg. 1.

*Proof.* We initiate our discussion with a single content scenario and subsequently extend our proof to a multi-video content context. Besides focusing on a particular ED, we simplify our notation by omitting the subscript  $e$ , provided there is no risk of confusion.

For any given input sequence  $\kappa$ , we define  $\text{PPVF}_i(\kappa) = \sum_{k \in \mathcal{S}_i} \lambda_i^k$  and  $\text{OPT}_i(\kappa) = \sum_{k \in \mathcal{S}_i^*} \lambda_i^k$  as the total utilities accrued by the PPVF algorithm (as outlined in Alg. 1) and the offline optimum (denoted as OPT), respectively. Here,  $\mathcal{S}_i$  and  $\mathcal{S}_i^*$  represent the set of requests selected to video  $i$  from the input sequence by these two methods. Let  $\Gamma_i$  represent the fraction of video  $i$ 's budget consumed by PPVF.

Furthermore, we define

$$\Lambda_i = \sum_{k \in (\mathcal{S} \cap \mathcal{S}^*)} \lambda_i^k, \quad \Lambda'_i = \sum_{k \in (\mathcal{S} \setminus \mathcal{S}^*)} \lambda_i^k,$$

and

$$\Upsilon_i = \sum_{k \in (\mathcal{S} \cap \mathcal{S}^*)} \Theta(\gamma_i^k) \cdot \epsilon_i, \quad \Upsilon'_i = \sum_{k \in (\mathcal{S} \setminus \mathcal{S}^*)} \Theta(\gamma_i^k) \cdot \epsilon_i.$$

First, for each request  $k \in \mathcal{S}_i$ , the efficiency  $\lambda_i^k / \epsilon_i$  is at least  $\Theta(\gamma_i^k)$ , i.e.,  $\lambda_i^k > \Theta(\gamma_i^k) \cdot \epsilon_i$ , where  $\gamma_i^k$  denotes the fraction of privacy budget of video  $i$  accessed at that specific time. Rounding down the utility  $\lambda_i^k$  of each request  $k$  chosen by PPVF to  $\Theta(\gamma_i^k) \cdot \epsilon_i$ , we ascertain that

$$\Upsilon_i \leq \Lambda_i, \quad (19a)$$

$$\Upsilon'_i \leq \Lambda'_i. \quad (19b)$$

Recall that  $\Theta(\gamma)$  is a monotonically increasing function with respect to  $\gamma$ , we can also observe

$$\Upsilon_i \leq \Theta(\Gamma_i) \cdot E_i, \quad (20)$$

where  $E_i = \sum_{k \in (\mathcal{S} \cap \mathcal{S}^*)} \epsilon_i$ .

Continuing our analysis, for each request  $k \in \mathcal{S}_i^* - (\mathcal{S}_i \cap \mathcal{S}_i^*)$ , which represents requests selected by OPT but not by our PPVF algorithm, we have:

$$\lambda_i^k \leq \Theta(\gamma_i^k) \cdot \epsilon_i \leq \Theta(\Gamma_i) \cdot \epsilon_i. \quad (21)$$

Note that  $\xi - E_i$  is the remaining budget as per PPVF after selecting requests to set  $\mathcal{S} \cap \mathcal{S}^*$ , which represents the ideal maximum budget that OPT could employ to select the request to the set  $\mathcal{S}^* - \mathcal{S} \cap \mathcal{S}^*$ . Given the threshold function  $\Theta(\gamma)$  is monotonically increasing with respect to  $\gamma$ , we can derive:

$$\text{OPT}_i(\kappa) - \Lambda_i \leq \Theta(\Gamma_i) \cdot (\xi - E_i). \quad (22)$$

Since  $\text{PPVF}_i(\kappa) = \Lambda_i + \Lambda'_i$ , the above inequality implies that

$$\frac{\text{OPT}_i(\kappa)}{\text{PPVF}_i(\kappa)} \leq \frac{\Lambda_i + \Theta(\Gamma_i) \cdot (\xi - E_i)}{\Lambda_i + \Lambda'_i}. \quad (23)$$

Additionally, considering  $\text{OPT}_i(\kappa) \geq \text{PPVF}_i(\kappa)$ , we always have  $\Theta(\Gamma_i) \cdot (\xi - E_i) \geq \Lambda'_i$ . Thus, if we reduce  $\Lambda_i$  to  $\Upsilon_i$  in both denominator and numerator of Eq. (23), the ratio of

$\frac{\text{OPT}_i(\kappa)}{\text{PPVF}_i(\kappa)}$  increases. In conclusion, combining the inequations in Eqs. (19)-(23), we have:

$$\begin{aligned} \frac{\text{OPT}_i(\kappa)}{\text{PPVF}_i(\kappa)} &\leq \frac{\Upsilon_i + \Theta(\Gamma_i) \cdot (\xi - E_i)}{\Upsilon_i + \Lambda'_i} \\ &\leq \frac{\Theta(\Gamma_i) \cdot E_i + \Theta(\Gamma_i) \cdot (\xi - E_i)}{\Upsilon_i + \Lambda'_i} \\ &\leq \frac{\Theta(\Gamma_i) \cdot \xi}{\Upsilon_i + \Upsilon'_i} = \frac{\Theta(\Gamma_i)}{\sum_{k \in \mathcal{S}} \Theta(\gamma_i^k) \Delta \gamma_i^k}. \end{aligned} \quad (24)$$

Recall that  $\Gamma_e = \frac{1}{1 + \ln(U_e/L_e)}$  is the lowest threshold in function  $\Theta_e(\cdot)$  at any ED  $e$ . Based on Assumption 1, indicating that  $\Delta \gamma_{e,i}^k \rightarrow 0$ , we have:

$$\begin{aligned} \sum_{k \in \mathcal{S}_{e,i}} \Theta_e(\gamma_{e,i}^k) \Delta \gamma_{e,i}^k &\approx \int_0^{\Gamma_i} \Theta_e(\gamma) d\gamma \\ &= \int_0^{\Gamma_e} L_e \cdot d\gamma + \int_{\Gamma_e}^{\Gamma_i} \frac{L_e}{\exp(1)} \ln \left( \frac{U_e \cdot \exp(1)}{L_e} \right)^\gamma \cdot d\gamma \\ &= \Gamma_e \cdot \left( L_e + \frac{L_e}{\exp(1)} \left( \ln \left( \frac{U_e \cdot \exp(1)}{L_e} \right)^{\Gamma_i} - \ln \left( \frac{U_e \cdot \exp(1)}{L_e} \right)^{\Gamma_e} \right) \right) \\ &= \Gamma_e \cdot \frac{L_e}{\exp(1)} \ln \left( \frac{U_e \cdot \exp(1)}{L_e} \right)^{\Gamma_i} \\ &= \Gamma_e \cdot \Theta_e(\Gamma_i). \end{aligned} \quad (25)$$

Therefore, the ratio  $\frac{\text{OPT}_i(\kappa)}{\text{PPVF}_i(\kappa)}$  at ED  $e$  can be derived into

$$\frac{\text{OPT}_i(\kappa)}{\text{PPVF}_i(\kappa)} \leq \frac{\Theta(\Gamma_i)}{\Gamma_e \cdot \Theta(\Gamma_i)} = 1 + \ln(U_e/L_e). \quad (26)$$

Denote  $\mathcal{J}_e^*$  and  $\mathcal{J}_e^{\text{PPVF}}$  as the final sum of utility obtained by solution of offline optimum and PPVF, respectively. Following the proof in [28], [29], the CR of our online algorithm at any ED  $e$  can be obtained by by summing up all single video  $i$ :

$$\begin{aligned} \mathcal{J}_e^* &= \sum_i \text{OPT}_i(\kappa) \\ &\leq \sum_i (1 + \ln(U_e/L_e)) \cdot \text{PPVF}_i(\kappa) \\ &= (1 + \ln(U_e/L_e)) \cdot \mathcal{J}_e^{\text{PPVF}}. \end{aligned} \quad (27)$$

To sum up, we can similarly prove that Alg. 1 achieves the best competition ratio (CR) among all online solutions under Assumption 1.

$$CR = \max_{\kappa} \frac{\mathcal{J}_e^*}{\mathcal{J}_e^{\text{PPVF}}} = (1 + \ln(U_e/L_e)). \quad (28)$$

Proof completes.  $\square$

## APPENDIX B

### DETAILED DERIVATION FOR LOG-LIKELIHOOD FUNCTION

Let the occurrence time  $t^\nu$  be the time of the last viewing request in the historical viewing request set  $\mathcal{V}_e^\nu$ . Given the overall intensity function  $\hat{h}'_e(t) = \sum_{i \in \mathcal{V}_e} \hat{h}'_e(t, i)$  for any ED  $e$ , the probability that no request occurs in the period  $[t^\nu, t)$ ,  $t < t^{\nu+1}$  is

$$P \{ \text{no request occurs in } [t^\nu, t) \mid \mathcal{V}_e \} = \exp \left[ - \int_{t^\nu}^t \hat{h}'_e(t') dt' \right].$$



TABLE III: CHR (%) under different caching capacity  $c_e$ . SAGE and BESTFIT are implemented with our video utility predictor.

Caching Capacity	0.1%	0.25%	0.5%	0.75%	1%	2.5%	5%	7.5%	10%
PPVF	<b>3.782</b>	<b>7.270</b>	<b>11.728</b>	<b>15.764</b>	<b>19.492</b>	35.487	47.936	55.505	60.677
SAGE	3.768	7.266	11.719	15.761	19.491	35.431	47.905	55.451	60.569
BESTFIT	3.779	7.256	11.710	15.688	19.431	<b>35.762</b>	<b>48.394</b>	<b>55.971</b>	<b>61.528</b>
HRS	3.033	6.082	9.974	13.336	16.327	29.632	41.700	48.845	52.483
MAV	3.029	5.740	9.079	11.737	14.083	25.209	38.351	47.365	53.862
LRU	1.293	2.965	5.539	7.989	10.257	21.739	35.366	45.367	52.436
LFU	1.256	2.878	5.847	8.506	10.925	23.542	38.312	48.171	54.896

**Algorithm 3:** Online and distributed parameters learning algorithm for MEP.

**Input:** Online update time set  $\mathcal{Q}$ , Update interval  $\Delta t$   
**Output:**  $\theta$

```

1 for  $t_\theta \in \mathcal{Q}$  do
2    $n \leftarrow 0$  and initialize  $\theta^{(n)}$  with the outdated parameters;
3   while The termination condition is not satisfied do
4      $l_G \leftarrow 0, \nabla_G \leftarrow [0]^{(2 \times D+1) \times I};$ 
5     for  $\forall e \in \mathcal{E}$  do
6        $l_e, \nabla_e \leftarrow \text{EDLocalTraining}(t_\theta, \Delta t, \theta^{(n)});$ 
7        $l_G \leftarrow l_G + l_e, \nabla_G \leftarrow \nabla_G + \nabla_e;$ 
8     end
9     Update  $\theta^{(n+1)} \leftarrow \theta^{(n)}$  with  $l_G, \nabla_G$  and the penalty term  $\rho;$ 
10     $n \leftarrow n + 1;$ 
11  end
12 end
13 return  $\theta$ 
14 Function EDLocalTraining( $t_\theta, \Delta t, \theta^{(n)}$ ):
15    $l_e \leftarrow 0, \nabla_\beta \leftarrow [0]^I, \nabla_p \leftarrow [0]^{I \times D}, \nabla_q \leftarrow [0]^{I \times D};$ 
16   for  $\forall i \in \mathcal{I}$  do
17     Collect the timestamp set of historical viewing requests  $\mathcal{T}_{e,i}^{t_\theta - \Delta t, t_\theta}$  from the local dataset;
18     for  $\forall \tau \in \mathcal{T}_{e,i}^{t_\theta - \Delta t, t_\theta}$  do
19       Obtain the intensity  $\hat{h}_e(i, \tau)$  by Eq. (10);
20        $l_e \leftarrow l_e + \log \hat{h}_e(i, \tau);$ 
21     end
22     Calculate the term of integration  $l' = \int_{t_\theta - \Delta t}^{t_\theta} \hat{h}_e(i, t) dt$  in Eq.(12);
23      $l_e \leftarrow l_e - l';$ 
24     Calculate the gradient  $\nabla_\beta[i], \nabla_p[i]$  by Eqs. (14) and (13), respectively;
25     for  $\forall j \in \mathcal{I}$  do
26       Calculate the gradient  $\nabla_q[j]$  by Eq. (15);
27     end
28   end
29   Combine  $\nabla_\beta, \nabla_p[i], \nabla_q[j]$  to gain the whole  $\nabla_e;$ 
30   return  $l_e, \nabla_e$ 

```

time  $t$  is given by

$$P\{i, t \mid \mathcal{V}_e^\nu\} = \hat{h}_e(t, i) \exp \left[ - \int_{t^\nu}^t \hat{h}'_e(t') dt' \right].$$

For convenience, we let  $t^0 = 0$  and align the all-time series for different videos  $i$  to the same initial point. Recall that  $\mathcal{V}_e$  as the historical dataset of all viewing requests at ED  $e$  between the time  $(0, T]$ , it is easy to derive the likelihood function for all the parameters  $\theta \stackrel{\text{def}}{=} \{p, q, \beta\}$  shown in Eq. (29).

$$\begin{aligned}
l_e(\theta \mid \mathcal{V}_e) &= \prod_{\nu=1}^{|\mathcal{V}_e|} \hat{h}_e(i, t^\nu) \exp \left[ - \int_{t^{\nu-1}}^{t^\nu} \hat{h}'_e(t) dt \right] \\
&\quad \cdot \exp \left[ - \int_{t \mid \mathcal{V}_e}^T \hat{h}'_e(t) dt \right], \\
&= \prod_{i \in \mathcal{I}} \prod_{\tau \in \mathcal{T}_{e,i}} \hat{h}_e(i, \tau) \cdot \exp \left[ - \int_0^T \hat{h}_e(t) dt \right].
\end{aligned} \tag{29}$$

where  $\mathcal{T}_{e,i}$  denoted the timestamp set of the viewing requests to video  $i$  arriving at ED  $e$  in  $[0, T]$ . In order to facilitate calculation, we can further derive the log-likelihood function to optimize all the parameters  $\theta$ , which is defined as:

$$ll_e(\theta \mid \mathcal{V}_e) = \sum_{i \in \mathcal{I}} \sum_{\tau \in \mathcal{T}_{e,i}} \log \hat{h}_e(i, \tau) - \int_0^T \hat{h}_e(i, t') dt'. \tag{30}$$

## APPENDIX C

### FL-BASED ONLINE PARAMETERS ESTIMATION ALGORITHM

The detailed algorithm for online FL-based parameter estimation for the MEP model is presented in Alg. 3.

## APPENDIX D

### SUPPLEMENT OF CHR RESULTS

To better display the caching performance difference among different baselines, we further present numerical CHR results in Table III. Table III demonstrates that PPVF / SAGE / BESTFIT consistently attain the highest CHR compared to other caching baselines due to our efficient utility prediction algorithm. PPVF is slightly better than SAGE and BESTFIT in terms of the CHR performance among the small capacities, while BESTFIT achieves the highest CHR when the caching capacity is large.

Thus, the probability that a request for video  $i$  occurs at