Decentralized Fairness Aware Multi Task Federated Learning for VR Network

Krishnendu S. Tharakan, *Member, IEEE*, Carlo Fischione, *Fellow, IEEE*
School of Electrical Engineering and Computer Science, KTH Royal Institute of Technology, Stockholm, Sweden
Email: {tharakan, carlofi}@kth.se

***Abstract*—Wireless connectivity promises to unshackle virtual reality (VR) experiences, allowing users to engage from anywhere, anytime. However, delivering seamless, high-quality, real-time VR video wirelessly is challenging due to the stringent quality of experience requirements, low latency constraints, and limited VR device capabilities. This paper addresses these challenges by introducing a novel decentralized multi task fair federated learning (DMTFL) based caching that caches and prefetches each VR user's field of view (FOV) at base stations (BSs) based on the caching strategies tailored to each BS. In federated learning (FL) in its naive form, often biases toward certain users, and a single global model fails to capture the statistical heterogeneity across users and BSs. In contrast, the proposed DMTFL algorithm personalizes content delivery by learning individual caching models at each BS. These models are further optimized to perform well under any target distribution, while providing theoretical guarantees via Rademacher complexity and a probably approximately correct (PAC) bound on the loss. Using a realistic VR head-tracking dataset, our simulations demonstrate the superiority of our proposed DMTFL algorithm compared to baseline algorithms.**

***Index Terms*—Virtual reality, decentralized learning, fairness, multi task federated learning.**

I. Introduction

Virtual reality (VR) technology is poised to transform the way users engage with their surroundings, with the global VR market expected to exceed 435 billion USD by 2030 [1]. Realizing this potential, however, hinges on addressing a major challenge, which is ensuring reliable connectivity and high-quality user experiences over inherently unstable wireless channels [2].

VR applications heavily depend on 360° immersive video technology to create compelling user experiences [3]. While current video coding techniques often tailor streaming to user attention and field of view (FOV) [4], delivering real-time, tile-based FOV content remains a major challenge. The steps involved in detecting, retrieving, and transmitting the appropriate tiles for a user's FOV can introduce considerable latency, an issue that worsens as the number of users grows. This latency can critically degrade the user experience, making it hard to meet stringent delay requirements. Among them, proactive caching of VR content at the network edge stands out as a particularly promising solution to improve delivery efficiency and reduce latency [5], [6].

Much of the existing research on VR content delivery concentrates on enhancing the throughput of wireless VR networks through advanced wireless resource allocation methods. For instance, the work in [7], proposes an iterative algorithm for optimizing wireless transmission in multiplayer interactive VR gaming frameworks using mobile edge computing (MEC), which employs a truncated first-order Taylor expansion of the objective function, which is progressively refined through iterative updates. Similarly, [8] introduces a two-phase method for associating VR users with mmWave access points, the first phase applies a graph-theoretic assignment technique, followed by optimization using geometric programming.

To the best of our knowledge, leveraging a decentralized multi-task fair federated learning (DMTFL) framework for VR content delivery while explicitly accounting for the users' FOV requests, and promoting fairness among users has not yet been explored within the wireless communications domain. Motivated by this gap, we tackle the caching problem through a decentralized learning framework, without relying on prior statistical assumptions about the request sequence. This paper proposes a novel wireless VR network architecture empowered by MEC. Our proposed DMTFL algorithm enables personalized caching strategies at each BS, ensuring equitable service quality and robustness even under non-independent and identically distributed (non-iid) user request patterns.

To this end, we present a unified framework with rigorous theoretical guarantees and introduce a DMTFL based caching algorithm. The fairness aspect ensures that caching strategies are not biased toward particular BSs or user distributions, enabling equitable service quality across the network. Meanwhile, the multi-task aspect allows the proposed DMTFL algorithm to learn separate caching models for each BS, thereby providing personalized and locally optimized strategies tailored to the data and user demands at each BS. Further, to theoretically justify the performance of the proposed decentralized caching strategies, we incorporate tools from statistical learning theory, specifically the probably approximately correct (PAC) framework.

Furthermore, we use Rademacher complexity as a measure of function class capacity, quantifying the ability of our learning algorithm to generalize beyond the observed samples. This approach enables us to formally establish a bound (Theorem 1) that ensures the proposed algorithm not only performs well on the training data but also offers reliable guarantees on future requests, which is crucial for robust VR content delivery. Simulation results demonstrate that the proposed algorithm can substantially improve communication

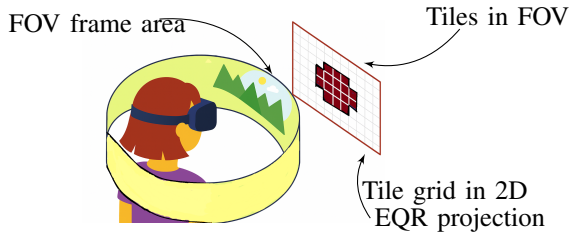

Fig. 1: Tiled FOV representation based on a user's FOV within the equirectangular (EQR) projection of a 360° VR scene

performance.

II. System Model and Problem Formulation

We consider B BSs denoted by the set $\mathcal{B} = \{1, 2, \ldots, b, \ldots, B\}$ and U users denoted by the set $\mathcal{U} = \{1, 2, \ldots, i, \ldots, U\}$. In this decentralized setting, BSs communicate and exchange models through limited-capacity links. Following a similar approach to [9], we assume that only a portion of the full 360° VR video is requested by a user referred to as the FOV. To facilitate processing, the 360° video is projected onto a two-dimensional plane and divided into $F = N \times P$ tiles. The user requests a subset of tiles from the set $\mathcal{F} = \{1, 2, \ldots, f, \ldots, F\}$, as illustrated in Fig. 1.
We denote the set of users connected to the b-th BS as $\mathcal{U}_b$. Let the feature vector for the b-th BS for the f-th FOV (eg. past demand, FoV history, etc.) be denoted as $x_{b,f}$. The total demand for the f-th FOV at the b-th BS is given by $y_{b,f}$, where $y_{b,f}$ serves as the target label. To evaluate the VR video quality, we use the mean squared error (MSE) metric. Let $c_{b,f}$ be the binary indicator representing the f-th cached FOV at the b-th BS, and define the MSE for the f-th FOV from the b-th BS as $e_{b,f} = (c_{b,f} - y_{b,f})^2$. Furthermore, to account for data heterogeneity, the data across devices is assumed to be non-iid, and are drawn from an underlying distribution $\mathcal{D}_b$ associated with the b-th BS, with each $\mathcal{D}_b = \big((x_{b,i}, y_{b,i}), \ldots, (x_{b,m_b}, y_{b,m_b})\big)$ of size m_b, which which captures the local data at the b-th BS, including user FOV requests. Let $\boldsymbol{\mathcal{D}} = \{\mathcal{D}_1, \ldots, \mathcal{D}_B\}$ represent the set of all distributions of the BSs.

In this work, we propose a novel algorithm, which we call DMTFL, where each BS learns a personalized caching vector $\boldsymbol{\phi_b}$ by minimizing its local empirical loss while also incorporating information from neighboring BSs to promote fairness and avoid bias toward any particular BS. Formally, each BS employ a DMTFL based caching strategy given by $\boldsymbol{\phi}_b$, where $\boldsymbol{\phi}_b = [\phi_{b,1}, \phi_{b,2}, \ldots, \phi_{b,F}]$ denotes the caching strategy employed at the b-th BS for the f-th FOV. Let $\Phi = \{\boldsymbol{\phi}_1, \boldsymbol{\phi}_2, \ldots, \boldsymbol{\phi}_B\} \subset \mathbb{R}^{F \times B}$ be the set of all caching strategies. The effectiveness of the caching scheme is evaluated using the following loss function:

$$\mathcal{L}_b(\boldsymbol{\phi}_b) = \mathbb{E}_{(x,y)\in\mathcal{D}_b} \sum_{f\in\mathcal{F}} \phi_{b,f} \log_{10}(e_{b,f}). \tag{1}$$

For the ease of notation, we denote $\mathbb{E}_{(x,y)\in\mathcal{D}_b} \sum_{f\in\mathcal{F}} \phi_{b,f} \log_{10}(e_{b,f}) = \mathbb{E}_{(x,y)\in\mathcal{D}_b} \ell(h_{\boldsymbol{\phi}}(\boldsymbol{x}), \boldsymbol{y})$, where ℓ is the loss function that combines the caching strategy and the mean squared error, and $h_{\boldsymbol{\phi}}(\boldsymbol{x})$ represents the caching policy predicted indicator vector for the given features.

Our primary goal is to evaluate the performance of the proposed algorithm by minimizing the loss function. The formal definition of the DMTFL algorithm is given later in Section III. It is important to recognize that employing a naive federated learning (FL) approach typically leads to learning a single, shared caching strategy across all BSs (i.e., $\boldsymbol{\phi}_b$ would be same across the BSs). While FL offers the advantage of leveraging local datasets and computation at edge BSs with minimal communication overhead, it also introduces specific challenges. In its standard form such as FedAvg [10] or FedProx [11], classical FL aims to train a single global model using data distributed across numerous, potentially millions of remote devices. This approach often overlooks the unique characteristics and contributions of individual BSs. When models are trained independently at different BSs with heterogeneous data distributions, the resulting statistical relationships may vary significantly. As a result, aggregating these into a single global model can lead to suboptimal performance and reduced inference accuracy.

Another key problem in FL is that of fairness. In many practical cases, the resulting learning models can exhibit bias or lack fairness [12]. A key reason these concerns arise is that the target distribution for which the FL model is learned is often unspecified. To account for fairness, we consider an agnostic FL setting, where the caching strategy is optimized for any possible target distribution formed by a mixture of the BS distributions. Thus, to address these limitations, our work adopts a personalized FL (also known as multi-task FL) as well as fair FL (also known as agnostic FL) framework that learns separate models for each BS, explicitly accounting for any target distribution in the federated setting. Instead of tailoring the caching strategy to a particular distribution, which risks misalignment with the target, we formulate an agnostic and more robust objective. In this setting, the target distribution is modeled as an unknown mixture of the distributions $\mathcal{D}_b, b = 1, \ldots, B$, that is $\mathcal{D}_w = \sum_{b=1}^{B} w_b \mathcal{D}_b$ for some $w \in \Delta_B$, or any w in a subset $\Lambda \in \Delta_B$, where Δ_B stands for the simplex over B. Given that the mixture weight w is unknown, we must design a solution that is robust across all possible w values within the simplex. This consideration naturally leads to the following optimization problem.

$$\min_{\boldsymbol{\phi}_1, \boldsymbol{\phi}_2, \ldots, \boldsymbol{\phi}_B} \sup_{\mathbf{w}\in\Lambda} \left\{ \theta_{\Phi,\mathbf{w}} = \sum_{b=1}^{B} w_b \mathcal{L}_b(\boldsymbol{\phi}_b, \mathcal{D}_b) \right\}, \tag{2}$$

where the coefficients satisfy $\sum_b^B w_b = 1$, and $\mathbf{w} = (w_1, \ldots, w_B)$. Note that to address the problem above, each BS first estimates its local loss $\mathcal{L}_b(\boldsymbol{\phi}_b)$ using the empirical average $\hat{\mathcal{L}}_b(\boldsymbol{\phi}_b, \mathcal{D}_b)$. This estimate relies entirely on the BS's own data, which, if limited, can be quite noisy and lead to poor performance. A practical way to mitigate this issue is to also incorporate information from neighboring BSs, particularly when their data distributions are similar. In the ideal case of iid data across devices, simply averaging all estimates works well. However, in non-iid scenarios, treating all neighbors

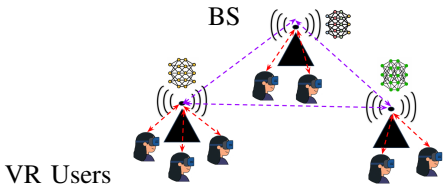

Fig. 2: Decentralized multi-task fair federated learning

equally can degrade the accuracy of the loss estimate. A more refined approach is to assign carefully chosen weights to each BS's data, allowing the loss estimate to better reflect the true local distribution. Once these optimal weights are determined, the overall model becomes significantly more robust and personalized. The caching strategy can be personalized by assigning distinct weights to each BS, as demonstrated in [13]. This results in the following optimization problem, which is solved in a distributed manner.

$$\min_{\boldsymbol{\phi}_1,\ldots,\boldsymbol{\phi}_B} \sup_{\mathbf{w}\in\Lambda} \left\{ \widehat{\theta}_{\Phi,\mathbf{w},\boldsymbol{\alpha}}(\boldsymbol{\mathcal{D}}) = \sum_{b=1}^{B} w_b \sum_{i=1}^{B} \alpha_{b,i} \hat{\mathcal{L}}_b(\boldsymbol{\phi}_i, \mathcal{D}_b) \right\}, \tag{3}$$

where the coefficients satisfy $\sum_i^B \alpha_{b,i} = 1$, and $\boldsymbol{\alpha} = \{\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_B\}$, and $\boldsymbol{\alpha}_b = \{\alpha_{b,1}, \ldots, \alpha_{b,B}\}$. Our approach explicitly accounts for spatial heterogeneity across devices and further incorporates this heterogeneity into the weighting of each loss metric. The main contribution of this paper is to derive a PAC bound on the difference $\theta_{\Phi,\mathbf{w}} - \widehat{\theta}_{\Phi,\mathbf{w},\boldsymbol{\alpha}}(\boldsymbol{\mathcal{S}})$. The subsequent section delves into the details of this distributed & fair multi-task FL based caching strategy.

III. DMTFL Algorithm

Drawing on distributed and multi-task learning principles, we introduce a decentralized multi-task structure where each BS learns a personalized caching vector that accounts for both local user demands and cross-BS fairness considerations. As illustrated in Fig. 2, we adopt a fair and multi-task FL framework, where each BS learns its own personalized caching model using local data. By modeling the target distribution as an unknown mixture and allowing separate models per BS, this approach effectively captures statistical heterogeneity across BSs. Consequently, it ensures that caching decisions are both tailored to local data characteristics and robust to distributional shifts, thereby promoting fairness and improving overall system efficiency. As depicted in Fig. 2, data collection across the network is non-iid, with variations in both data distribution and volume among BSs. In order to state the main result, we first introduce the following three key quantities: (i) Rademacher complexity, (ii) minimum ϵ-cover, and (iii) discrepancy, each defined below.

Definition 1. *(Minimax weighted Rademacher complexity): The Rademacher complexity for a given set of caching strategy Φ for a given $\mathbf{w} \in \Lambda$ is defined as*

$$\mathcal{R}_{\mathbf{w}}(\Phi) = \mathop{\mathbb{E}}_{\mathcal{D}_1,\ldots,\mathcal{D}_B} \left[\sup_{\substack{\boldsymbol{\phi}_b\in\boldsymbol{\Phi}, \\ \boldsymbol{\alpha}\in\Delta_B}} \sum_{b,i=1}^{B} \frac{w_b \alpha_{b,i}}{m_b} \sum_{j=1}^{m_b} \sigma_{bi,j}\, \ell\big(h_\phi(x_{b,j}), y_{b,j}\big) \right]$$

where $\mathcal{D}_b = \big((x_{b,i}, y_{b,i}), \ldots, (x_{b,m_b}, y_{b,m_b})\big)$ is a sample of size m_b and $\sigma_{bi,j}$ is a collection of Rademacher variables, that is uniformly distributed random variables taking values in $\{-1, 1\}$. max weighted Rademacher complexity is defined as $\mathcal{R}_\Lambda(\Phi) = \max_{\mathbf{w}\in\Lambda} \mathcal{R}_{\mathbf{w}}(\Phi)$.

Definition 2. *(Minimum ϵ cover): The set $\{c_1, \ldots, c_d\}$ is said to be an ϵ-cover of Λ with respect to the ℓ_1-distance if $\Lambda \subseteq \bigcup_{i=1}^{d} B(c_i, \epsilon)$, where the ℓ_1 ball is defined as $B(c_i, \epsilon) := \{z \in \Lambda : \|z - c_i\|_1 < \epsilon\}$. The minimum ϵ-cover, denoted by Λ_ϵ, is an ϵ-cover of Λ with the smallest possible cardinality d.*

Definition 3. *(Discrepancy): The discrepancy between two data distributions $\mathcal{D}_i$ and $\mathcal{D}_j$ corresponding to the BS i and j, respectively, measured using the loss function $l : \mathcal{Y}\times\mathcal{Y} \to \mathbb{R}^+$ is defined as $v_{i,j} = \sup_{\boldsymbol{\phi}} |\mathcal{L}_i(\boldsymbol{\phi}) - \mathcal{L}_j(\boldsymbol{\phi})|$*

The first term above represents the average loss at the ith BS, while the second term corresponds to the average loss at the jth BS. Their difference quantifies how distinct the two data distributions are with respect to the parameter $\boldsymbol{\phi}$. By maximizing over $\boldsymbol{\phi}$, we obtain the worst-case difference. When the data at the two BSs are iid, the discrepancy is zero. Conversely, if the data distributions differ significantly, the discrepancy will be large. A straightforward approach to incorporate this into an algorithm is to assign higher weights α when the discrepancy is small. In what follows, we present a theoretical result that offers a principled method for selecting these weights systematically. Specifically, the following theorem establishes a PAC bound on the difference between the true average loss in (2) and its empirical estimate in (3), leveraging the definitions introduced above. By providing these theoretical guarantees, our proposed algorithm offers both practical benefits and a deeper understanding of its effectiveness in handling dynamic FOV request patterns.

A. Theoretical Guarantees & Proposed Algorithm

In this subsection, we present a high probability bound on the performance of the proposed DMTFL algorithm. This analysis offers insights into selecting appropriate weights and optimizing the sequence of caching policies over time.

Theorem 1. *(PAC Bound) Assuming the loss function is bounded, i.e., $\ell(a, b) < H, \forall a, b \in \mathcal{Y}$ then for any $\epsilon > 0$, with probability at least $1 - \delta$, $\delta > 0$, the following holds:*

$$\theta_{\Phi,\mathbf{w}} \le \widehat{\theta}_{\Phi,\mathbf{w},\boldsymbol{\alpha}}(\boldsymbol{\mathcal{D}}) + 2\mathcal{R}_\Lambda(\Phi) + H\,\mathcal{P}(\mathbf{w}, \boldsymbol{\alpha}) + HB\epsilon, \tag{4}$$

where $\mathcal{P}(\mathbf{w}, \boldsymbol{\alpha}) = \sqrt{\frac{1}{2}\sum_{b=1}^{B}\sum_{i=1}^{B}\left(\frac{w_b\alpha_{b,i}}{m_b}\right)^2 \log\left(\frac{|\Lambda_\epsilon|}{\delta}\right)}$ $+ \frac{1}{H}\sum_{b=1}^{B}\sum_{i=1}^{B} w_b\alpha_{b,i}v_{bi}$, v_{bi} denotes the discrepancy, and Λ_ϵ is the minimum ϵ-cover of Λ.

Proof: See Appendix A. ∎

Remark 1. *The covering parameter ϵ can be chosen moderately (e.g., 0.1), since the PAC bound depends only logarithmically on $|\Lambda_\epsilon|$, ensuring practical feasibility even for large B.*

Theorem 1 establishes a high-probability lower bound on the term $\theta_{\Phi,\mathbf{w}} - \widehat{\theta}_{\Phi,\mathbf{w},\boldsymbol{\alpha}}(\boldsymbol{\mathcal{D}})$. Our goal is to optimize the caching

strategy, by carefully selecting the coefficients $\alpha_{b,i}$, and $\boldsymbol{\phi}_b$, such that left hand term in Theorem 1 is minimized. To achieve this minimization, we solve the following problem as shown below

$$\min_{\substack{\boldsymbol{\phi}_1,\ldots,\boldsymbol{\phi}_B \\ \boldsymbol{\alpha}}} \Bigg\{ \widehat{\Theta}_{\Phi,\mathbf{w},\boldsymbol{\alpha}} = \widehat{\theta}_{\Phi,\mathbf{w},\boldsymbol{\alpha}}(\boldsymbol{\mathcal{D}}) + \sum_{b=1}^{B} \rho_b ||\boldsymbol{\phi}_b||_2 + H\,\mathcal{P}(\mathbf{w},\boldsymbol{\alpha}) \Bigg\} \quad (5)$$

Algorithm 1 DMTFL Algorithm

procedure PROPOSED DMTFL
  Initialize $v_{b,i}=1$, $\boldsymbol{\phi}_b^0$, $\alpha_{b,i}^0$, $\forall\, b,i$
  for $t=1,2,\ldots,T$ **do**
    for $b=1,2,\ldots,B$ **do**
      for $i=1,2,\ldots,B$ **do**
        for $n=1,2,\ldots,N$ **do** and $b\neq i$
          Broadcast $\boldsymbol{\phi}_b^n$ to devices b and i
          Obtain sub-gradients, and losses from the BS b and i respectively
          Update $v_{b,i}^n$

$$v_{b,i}^n = |\widehat{\mathcal{L}}(\boldsymbol{\phi}_b^n,\mathcal{D}_b) - \widehat{\mathcal{L}}(\boldsymbol{\phi}_b^n,\mathcal{D}_i)|$$

          Gradient ascent on $\boldsymbol{\phi}_b$

$$\boldsymbol{\phi}_b^{n+1} = \boldsymbol{\phi}_b^n + \mu \partial_{\boldsymbol{\phi}}(\Delta_{b,i}(\boldsymbol{\phi}_b^n))$$

        end for
      end for
      Gradient descent on $\boldsymbol{\phi}_b$

$$\boldsymbol{\phi}_b^{t+1} = \boldsymbol{\phi}_b^t - \eta \nabla_{\boldsymbol{\phi}_b} \widehat{\Theta}_{\Phi^t,\mathbf{w},\boldsymbol{\alpha}^t}$$

      Gradient descent with projection step on $\boldsymbol{\alpha}$

$$\alpha_b^{t+1} \leftarrow \Pi_{\Delta_B}\Big(\alpha_b^t - \upsilon_t \nabla_{\alpha_b} \widehat{\Theta}_{\Phi^{t+1},\mathbf{w},\boldsymbol{\alpha}^t}\Big)$$

    end for
    Broadcast $\boldsymbol{\phi}_b$ to all BSs
  end for
  Output final $\{\boldsymbol{\phi}_b^T\}_{b=1}^B$ and $\{\alpha_{b,i}^T\}_{b,i=1}^B$
end procedure

Note that the above minimization problem requires knowledge of the discrepancy values, but since each BS only has access to its local data, these discrepancies must be estimated collaboratively in a distributed setup. Thus, the minimization problem described in (5) must be addressed in a distributed manner, with minimal communication overhead between nodes. To enable this, BSs must approximate the discrepancies and use these estimates as proxies when designing the proposed DMTFL algorithm. As the discrepany terms are not directly observable, we utilize their estimates instead. To obtain these estimates, we employ a distributed gradient ascent algorithm, the steps of which are summarized in Algorithm 1. It is noteworthy to mention that the described implementation of DMTFL (Algorithm 1), provides the valuable flexibility to compute the estimates of individual caching strategies in a distributed manner across the network, with computations performed locally at each BS.

From the definition of the discrepancy the proposed DMTFL algorithm essentially requires to solve

$$v_{i,j} = \sup_{\boldsymbol{\phi}} |\mathcal{L}_i(\boldsymbol{\phi}) - \mathcal{L}_j(\boldsymbol{\phi})|, \quad (6)$$

for all the BSs. Since the true expected losses $\mathcal{L}_i(\boldsymbol{\phi})$ and $\mathcal{L}_j(\boldsymbol{\phi})$ are not directly accessible, they are approximated using their estimates denoted by $\widehat{\mathcal{L}}_i(\boldsymbol{\phi})$ and $\widehat{\mathcal{L}}_j(\boldsymbol{\phi})$ respectively. A natural approach to tackling this maximization problem is to apply a distributed gradient ascent algorithm. Algorithm 1 outlines the distributed implementation of this approach to estimate the discrepancies across the BSs. However, the presence of the absolute value in the discrepancy definition means that the gradient does not exist at all points. To overcome this issue, a generalized subgradient is used instead of the standard gradient as shown below.

$$\partial_{\boldsymbol{\phi}}(\Delta_{b,i}(\boldsymbol{\phi})) = \big(\nabla\widehat{\mathcal{L}}(\boldsymbol{\phi},\mathcal{D}_b) - \nabla\widehat{\mathcal{L}}(\boldsymbol{\phi},\mathcal{D}_i)\big)\, \mathbb{1}\{\widehat{\mathcal{L}}(\boldsymbol{\phi},\mathcal{D}_b) > \widehat{\mathcal{L}}(\boldsymbol{\phi},\mathcal{D}_i)\} - \big(\nabla\widehat{\mathcal{L}}(\boldsymbol{\phi},\mathcal{D}_b) - \nabla\widehat{\mathcal{L}}(\boldsymbol{\phi},\mathcal{D}_i)\big)\mathbb{1}\{\widehat{\mathcal{L}}(\boldsymbol{\phi},\mathcal{D}_b) \leq \widehat{\mathcal{L}}(\boldsymbol{\phi},\mathcal{D}_i)\}.$$

Since the algorithm employs a subgradient ascent approach, convergence to a local maximum is ensured under standard assumptions, following classical results in non-convex optimization [14]. By leveraging this subgradient approach, each of the BS can iteratively refine its local estimate without requiring exact gradients, thus enabling an efficient and communication-friendly distributed estimation procedure.

IV. Simulation Results

To validate the performance of the proposed algorithms, datasets from [15] are utilized. The dataset comprises tracked head movements of 50 users while watching a catalog of 10 HD 360° YouTube videos. A $100^\circ \times 100^\circ$ FOV is considered, and to build the tiled-FOV, the equirectangular projection of each of the video frames is divided into $N \times P$ tiles. The proposed DMTFL-based algorithm is compared with the following benchmark methods:

- FedAvg Algorithm: The central BS gathers the gradients from each BS's local loss to train a global model that minimizes the overall loss across all BSs [10].
- FedProx Algorithm: FedProx algorithm extends FedAvg by adding a proximal term to the local objective at each BS [11].
- Heuristic Caching: In this caching algorithm, it selects and caches the tiles with the highest overall global request frequency (i.e., most popular tiles across all users), regardless of individual user or BS preferences.

Fig. 3 illustrates the impact of the cache size of the BS on the average cache hit. As depicted in the figure, a clear trend emerges, indicating that as the cache size of the BS increases, the average cache hit naturally increases. It is clear from Fig. 3 that the proposed algorithm performs better than the baseline algorithms, demonstrating the benefit of using the proposed algorithm. For instance, the average cache hit for the algorithm is at least 55% higher as compared to the benchmark algorithms.

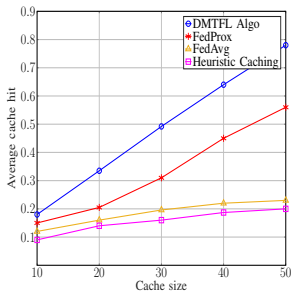

Fig. 3: Average cache hit versus cache size.

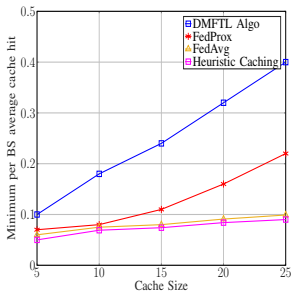

Fig. 4: Minimum average cache hit versus cache size.

Fig. 4 illustrates the minimum per-BS cache hit rate, which serves as a fairness indicator by highlighting the worst-served BS in the network. Unlike average or aggregate metrics, this measure directly captures the performance experienced by the most disadvantaged BS. A higher minimum hit rate indicates that the algorithm is effectively avoiding bias toward data-rich or majority BSs, aligning with the FL fairness objective of providing balanced and equitable performance across all BSs. As seen in the Fig. 4, the proposed DMTFL algorithm achieves a significantly higher minimum per-BS hit rate compared to baseline methods, demonstrating its superior fairness properties.

V. Conclusion

This paper tackles the critical challenge of VR content caching in highly dynamic and adversarial environments, leveraging the power of the FL framework. By explicitly addressing the challenges of statistical heterogeneity and fairness across BSs, the proposed approach enables personalized caching strategies tailored to each user's FOV demands. Furthermore, we established rigorous theoretical foundations through Rademacher complexity analysis and derived a PAC bound on the loss given by Theorem 1, providing strong generalization guarantees. The numerical results demonstrate the superior performance of our proposed algorithm compared to existing baseline methods, highlighting its effectiveness in handling future demands of dynamic VR content delivery. While this paper demonstrates the potential of the proposed DMTFL algorithm, several important aspects remain open for future investigation. Evaluating communication and computation overhead in practical MEC environments, especially as the number of BSs increases, is also a crucial direction. Furthermore, the proposed framework can also be extended to emerging music metaverse applications, where immersive VR environments are further enriched with dynamic music content. Finally, extending the framework to account for dynamic user mobility and BS associations will be considered in future work.

Acknowledgment

We acknowledge the support of the MUSMET project funded by the EIC Pathfinder Open scheme of the European Commission (grant agreement no. 101184379).

Appendix A
Proof of Theorem 1

Consider $\mathcal{D}_b = \{(x_{b,1}, y_{b,1}), (x_{b,2}, y_{b,2}), \ldots, (x_{b,m_b}, y_{b,m_b})\}$, which represents the set of data samples at the BS b. Each BS b is assumed to have caching strategy $\boldsymbol{\phi}_b \in \mathbb{R}^F$, for $b = 1, 2, \ldots, B$. Let $\Phi = \{\boldsymbol{\phi}_1, \boldsymbol{\phi}_2, \ldots, \boldsymbol{\phi}_B\} \subset \mathbb{R}^{F \times B}$ denote the collection of all caching strategies across the BSs. The weights $\boldsymbol{\phi}_b$ at the BS b are optimized using the loss function

$$L_b(\boldsymbol{\phi}_b) := \mathbb{E}_{(x_b, y_b) \sim \mathcal{D}_b}\big[l(h_{\boldsymbol{\phi}_b}(x_b), y_b)\big].$$

The weighted sum of the average losses across all the BS in the network is given by

$$\begin{aligned}
\theta_{\Phi, \boldsymbol{w}} &= \sum_{b=1}^{B} w_b \mathcal{L}_b(\boldsymbol{\phi}_b) \\
&\overset{(a)}{=} \sum_{b,i=1}^{B} w_b \alpha_{b,i} \mathcal{L}_i(\boldsymbol{\phi}_b) + \sum_{b,i=1}^{B} w_b \alpha_{b,i} \big[\mathcal{L}_b(\boldsymbol{\phi}_b) - \mathcal{L}_i(\boldsymbol{\phi}_b)\big], \\
&\overset{(b)}{=} \theta_{\Phi, \mathbf{w}, \boldsymbol{\alpha}} + \sum_{b=1}^{B} \sum_{i=1}^{B} w_b \alpha_{b,i} \big[\mathcal{L}_b(\boldsymbol{\phi}_b) - \mathcal{L}_i(\boldsymbol{\phi}_b)\big], \\
&\overset{(c)}{\leq} \theta_{\Phi, \mathbf{w}, \boldsymbol{\alpha}} + \sum_{b=1}^{B} \sum_{i=1}^{B} w_b \alpha_{b,i} v_{b,i}
\end{aligned}$$

where (a) follows from adding and subtracting the term $\sum_{b=1}^{B} \sum_{i=1}^{B} w_b \alpha_{b,i} \mathcal{L}_i(\phi_b)$, and using the fact that $\sum_{b=1}^{B} w_b = 1$, and $\sum_{i=1}^{B} \alpha_{b,i}^{B} = 1$, (b) follows from using the definition of $\theta_{\Phi, \mathbf{w}, \boldsymbol{\alpha}} = \sum_{b=1}^{B} \sum_{i=1}^{B} w_b \alpha_{b,i} \mathcal{L}_i(\phi_b)$, and (c) follows from the definition of discrepancy between two BSs b and i, i.e., $v_{b,i} = \sup_{\boldsymbol{\phi}} |\mathcal{L}_i(\boldsymbol{\phi}) - \mathcal{L}_j(\boldsymbol{\phi})|$. Thus, the above can be upper bounded as follows:

$$\theta_{\Phi, \mathbf{w}} \leq \theta_{\Phi, \mathbf{w}, \boldsymbol{\alpha}} + \sum_{b=1}^{B} \sum_{i=1}^{B} w_b \alpha_{b,i} v_{b,i}. \tag{7}$$

Note that $\widehat{\theta}_{\Phi, \mathbf{w}, \boldsymbol{\alpha}} = \sum_{b=1}^{B} w_b \sum_{i=1}^{B} \alpha_{b,i} \hat{\mathcal{L}}_b(\boldsymbol{\phi}_i, \mathcal{D}_b)$ is an estimate of the term $\theta_{\Phi, \mathbf{w}, \boldsymbol{\alpha}}$. Let $\Upsilon(\mathcal{D}) = \sup_{\boldsymbol{\phi} \in \Phi} \big(\theta_{\Phi, \mathbf{w}, \boldsymbol{\alpha}} - \widehat{\theta}_{\Phi, \mathbf{w}, \boldsymbol{\alpha}}(\mathcal{D})\big)$. Let us consider two sets of samples, $\mathcal{D}'_b = \{\mathcal{D}'_1, \ldots, \mathcal{D}'_B\}$ and $\mathcal{D}_b = \{\mathcal{D}_1, \ldots, \mathcal{D}_B\}$, which differ in exactly one data point. Specifically, let this differing element be $(x'_{b,i}, y'_{b,i}) \in \mathcal{D}'_b$ and $(x_{b,i}, y_{b,i}) \in \mathcal{D}_b$, where $b \in \{1, \ldots, B\}$ and $i \in \{1, \ldots, m_b\}$. To apply McDiarmid's inequality [14], it is necessary to bound the following difference:

$$\begin{aligned}
\Upsilon(\mathcal{D}') - \Upsilon(\mathcal{D}) &= \sup_{\boldsymbol{\phi} \in \Phi} \big(\theta_{\Phi, \mathbf{w}, \boldsymbol{\alpha}} - \widehat{\theta}_{\Phi, \mathbf{w}, \boldsymbol{\alpha}}(\mathcal{D}')\big) \\
&\quad - \sup_{\boldsymbol{\phi} \in \Phi} \big(\theta_{\Phi, \mathbf{w}, \boldsymbol{\alpha}} - \widehat{\theta}_{\Phi, \mathbf{w}, \boldsymbol{\alpha}}(\mathcal{D})\big) \\
&\overset{(a)}{\leq} \sup_{\boldsymbol{\phi} \in \Phi} \big[\big(\theta_{\Phi, \mathbf{w}, \boldsymbol{\alpha}} - \widehat{\theta}_{\Phi, \mathbf{w}, \boldsymbol{\alpha}}(\mathcal{D}')\big) - \big(\theta_{\Phi, \mathbf{w}, \boldsymbol{\alpha}} - \widehat{\theta}_{\Phi, \mathbf{w}, \boldsymbol{\alpha}}(\mathcal{D})\big)\big] \\
&\leq \sup_{\boldsymbol{\phi} \in \Phi} \big(\widehat{\theta}_{\Phi, \mathbf{w}, \boldsymbol{\alpha}}(\mathcal{D}) - \widehat{\theta}_{\Phi, \mathbf{w}, \boldsymbol{\alpha}}(\mathcal{D})'\big) \\
&\leq \sup_{\boldsymbol{\phi} \in \Phi} \left(\sum_{b,i=1}^{B} w_b \alpha_{b,i} \big[\hat{\mathcal{L}}_b(\boldsymbol{\phi}_i, \mathcal{D}_b) - \hat{\mathcal{L}}_b(\boldsymbol{\phi}_i, \mathcal{D}'_b)\big] \right) \\
&\overset{(b)}{\leq} \sum_{b,i=1}^{B} \frac{w_b}{m_i} \alpha_{b,i} H,
\end{aligned}$$

where (a) follows from the properties of the supremum, while (b) follows from the fact that the loss is assumed to be

bounded. Applying McDiarmid's inequality for any $\delta > 0$ and $\boldsymbol{\phi} \in \Phi$, it can be shown that the following holds with probability at least $1-\delta$,

$$\theta_{\Phi,\mathbf{w}} \leq \widehat{\theta}_{\Phi,\mathbf{w},\boldsymbol{\alpha}}(\boldsymbol{\mathcal{D}}) + \mathbb{E}_{\mathcal{D}}[\Upsilon(\mathcal{D})] + H\sqrt{\frac{1}{2}\sum_{i=1}^{B}\big(\sum_{b=1}^{B}\frac{w_b}{m_i}\alpha_{b,i}\big)^2\log(\frac{1}{\delta})}.$$

Therefore, by the union over Λ_ϵ, and by definition of Λ_ϵ, for any $\mathbf{w} \in \Lambda$, there exists $\mathbf{w}' \in \Lambda_\epsilon$ such that $\widehat{\theta}_{\Phi,\mathbf{w},\boldsymbol{\alpha}}(\boldsymbol{\mathcal{D}}) \leq \widehat{\theta}_{\Phi,\mathbf{w},\boldsymbol{\alpha}}(\boldsymbol{\mathcal{D}'}) + HB\epsilon$. In light of this, with probability of $1-\delta$, the following holds

$$\theta_{\Phi,\mathbf{w}} \leq \widehat{\theta}_{\Phi,\mathbf{w},\boldsymbol{\alpha}}(\boldsymbol{\mathcal{D}}) + \mathbb{E}_{\mathcal{D}}[\Upsilon(\mathcal{D})] + HB\epsilon + H\sqrt{\frac{1}{2}\sum_{i=1}^{B}\big(\sum_{b=1}^{B}\frac{w_b}{m_i}\alpha_{b,i}\big)^2\log(\frac{|\Lambda_\epsilon|}{\delta})}. \quad (8)$$

For the term $\mathbb{E}_{\mathcal{D}}[\Upsilon(\mathcal{D})]$, we have the following

$$\begin{aligned}
&\mathbb{E}_{\mathcal{D}}[\Upsilon(\mathcal{D})] = \mathop{\mathbb{E}}_{\mathcal{D}}\left[\sup_{\boldsymbol{\phi}\in\Phi}\left(\theta_{\Phi,\mathbf{w},\boldsymbol{\alpha}} - \widehat{\theta}_{\Phi,\mathbf{w},\boldsymbol{\alpha}}(\mathcal{D}')\right)\right] \\
&= \mathop{\mathbb{E}}_{\mathcal{D}}\left[\sup_{\boldsymbol{\phi}\in\Phi}\mathbb{E}_{\mathcal{D}'}\left(\widehat{\theta}_{\Phi,\mathbf{w},\boldsymbol{\alpha}}(\mathcal{D}') - \widehat{\theta}_{\Phi,\mathbf{w},\boldsymbol{\alpha}}(\mathcal{D})\right)\right] \\
&\overset{(a)}{\leq} \mathop{\mathbb{E}}_{\mathcal{D},\mathcal{D}'}\left[\sup_{\boldsymbol{\phi}\in\Phi}\left(\widehat{\theta}_{\Phi,\mathbf{w},\boldsymbol{\alpha}}(\mathcal{D}') - \widehat{\theta}_{\Phi,\mathbf{w},\boldsymbol{\alpha}}(\mathcal{D})\right)\right] \\
&\leq \mathop{\mathbb{E}}_{\mathcal{D},\mathcal{D}'}\left[\sup_{\boldsymbol{\phi}\in\Phi}\left(\sum_{b,i=1}^{B}\sum_{j=1}^{m_b}\frac{w_b\alpha_{b,i}}{m_j}[\ell(h_{\boldsymbol{\phi}}(x'_{b,i}), y'_{b,i}) \right.\right. \\
&\left.\left. -\ell(h_{\boldsymbol{\phi}}(x_{b,i}), y_{b,i})]\right)\right] \\
&\leq \mathop{\mathbb{E}}_{\mathcal{D},\mathcal{D}',\sigma}\left[\sup_{\boldsymbol{\phi}\in\Phi}\left(\sum_{b,i=1}^{B}\sum_{j=1}^{m_b}\frac{\sigma_{bi,j}w_b\alpha_{b,i}}{m_j}[\ell(h_{\boldsymbol{\phi}}(x'_{b,i}), y'_{b,i}) \right.\right. \\
&\left.\left. -\ell(h_{\boldsymbol{\phi}}(x_{b,i}), y_{b,i})]\right)\right] \\
&\overset{(b)}{\leq} \mathop{\mathbb{E}}_{\mathcal{D}',\sigma}\left[\sup_{\boldsymbol{\phi}\in\Phi}\left(\sum_{b,i=1}^{B}\sum_{j=1}^{m_b}\frac{\sigma_{bi,j}w_b\alpha_{b,i}}{m_j}[\ell(h_{\boldsymbol{\phi}}(x'_{b,i}), y'_{b,i})]\right)\right. \\
&\left. + \mathop{\mathbb{E}}_{\mathcal{D},\sigma}\left[\sup_{\boldsymbol{\phi}\in\Phi}\left(\sum_{b,i=1}^{B}\sum_{j=1}^{m_b}\frac{\sigma_{bi,j}w_b\alpha_{b,i}}{m_j}[-\ell(h_{\boldsymbol{\phi}}(x_{b,i}), y_{b,i})]\right)\right]\right. \\
&\leq 2\mathop{\mathbb{E}}_{\mathcal{D},\sigma}\left[\sup_{\boldsymbol{\phi}\in\Phi}\left(\sum_{b,i=1}^{B}\sum_{j=1}^{m_b}\frac{\sigma_{bi,j}w_b\alpha_{b,i}}{m_j}[\ell(h_{\boldsymbol{\phi}}(x_{b,i}), y_{b,i})]\right)\right] \\
&\leq 2\mathcal{R}_\Lambda(\Phi)
\end{aligned}$$

where (a) follows from the Jensen's inequality, and (b) follows from the standard Rademacher complexity upper bounds and from the definition of minimax weighted Rademacher complexity, and $\sigma_{bi,j}$ is the Rademacher random variable. Using the above result in 8, we get the following

$$\theta_{\Phi,\mathbf{w}} \leq \widehat{\theta}_{\Phi,\mathbf{w},\boldsymbol{\alpha}}(\boldsymbol{\mathcal{D}}) + 2\mathcal{R}_\Lambda(\Phi) + HB\epsilon + H\sqrt{\frac{1}{2}\sum_{i=1}^{B}\big(\sum_{b=1}^{B}\frac{w_b}{m_i}\alpha_{b,i}\big)^2\log(\frac{|\Lambda_\epsilon|}{\delta})}. \quad (9)$$

Next, substituting (9) in (7), we get the following:

$$\theta_{\Phi,\mathbf{w}} \leq \widehat{\theta}_{\Phi,\mathbf{w},\boldsymbol{\alpha}}(\boldsymbol{\mathcal{D}}) + 2\mathcal{R}_\Lambda(\Phi) + HB\epsilon + \sum_{b=1}^{B}\sum_{i=1}^{B}w_b\alpha_{b,i}v_{b,i} + H\sqrt{\frac{1}{2}\sum_{i=1}^{B}\big(\sum_{b=1}^{B}\frac{w_b}{m_i}\alpha_{b,i}\big)^2\log(\frac{|\Lambda_\epsilon|}{\delta})}. \quad (10)$$

This proves the theorem.

REFERENCES

[1] G. V. Research, "Virtual reality (VR) market size and share report, 2030," July 2023. [Online]. Available: https://www.grandviewresearch.com/industry-analysis/virtual-reality-vr-market

[2] F. Hu, Y. Deng, W. Saad, M. Bennis, and A. H. Aghvami, "Cellular-connected wireless virtual reality: Requirements, challenges, and solutions," *IEEE Communications Magazine*, vol. 58, no. 5, pp. 105–111, 2020.

[3] M. Chen, W. Saad, and C. Yin, "Virtual reality over wireless networks: Quality-of-service model and learning-based resource management," *IEEE Transactions on Communications*, vol. 66, no. 11, pp. 5621–5635, 2018.

[4] M. Zink, R. K. Sitaraman, and K. Nahrstedt, "Scalable 360° video stream delivery: Challenges, solutions, and opportunities," *Proceedings of the IEEE*, vol. 107, pp. 639–650, 2019.

[5] Y. Guo, Z. Qin, X. Tao, and G. Y. Li, "Federated multi-view synthesizing for metaverse," *IEEE Journal on Selected Areas in Communications*, vol. 42, no. 4, pp. 867–879, 2024.

[6] K. S. Tharakan, H. Dahrouj, N. Kouzayha, H. Elsawy, and T. Y. Al-Naffouri, "Personalized federated learning for cellular VR: Online learning and dynamic caching," *IEEE Transactions on Communications*, vol. 73, no. 8, pp. 6006–6022, 2025.

[7] Z. Chen, H. Zhu, L. Song, D. He, and B. Xia, "Wireless multiplayer interactive virtual reality game systems with edge computing: Modeling and optimization," *IEEE Transactions on Wireless Communications*, vol. 21, no. 11, pp. 9684–9699, 2022.

[8] S. Gupta, J. Chakareski, and P. Popovski, "mmWave networking and edge computing for scalable 360° video multi-user virtual reality," *IEEE Transactions on Image Processing*, vol. 32, pp. 377–391, 2023.

[9] P. Lungaro, R. Sjöberg, A. J. F. Valero, A. Mittal, and K. Tollmar, "Gaze-aware streaming solutions for the next generation of mobile VR experiences," *IEEE Transactions on Visualization and Computer Graphics*, vol. 24, no. 4, pp. 1535–1544, 2018.

[10] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y. Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, vol. 54, Apr 2017, pp. 1273–1282.

[11] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," in *Proceedings of Machine Learning and Systems*, 2020, pp. 429–450.

[12] M. Mohri, G. Sivek, and A. T. Suresh, "Agnostic federated learning," in *Proceedings of the 36th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, K. Chaudhuri and R. Salakhutdinov, Eds., vol. 97. PMLR, 09–15 Jun 2019, pp. 4615–4625.

[13] V. Smith, C.-K. Chiang, M. Sanjabi, and A. S. Talwalkar, "Federated multitask learning," in *Advances in Neural Information Processing Systems*, vol. 30, 2017, pp. 4424–4434.

[14] M. Mohri, A. Rostamizadeh, and A. Talwalkar, *Foundations of Machine Learning*. MIT press, 2018.

[15] W.-C. Lo, C.-L. Fan, J. Lee, C.-Y. Huang, K.-T. Chen, and C.-H. Hsu, "360° video viewing dataset in head-mounted virtual reality," in *Proceedings of the 8th ACM on Multimedia Systems Conference*, 2017, pp. 211–216.