Towards Neural Codec-Empowered 360° Video Streaming: A Saliency-aided Synergistic Approach

Jianxin Shi, Miao Zhang, Linfeng Shen, Jiangchuan Liu, *Fellow, IEEE*, Lingjun Pu, Jingdong Xu

***Abstract*—Networked 360° video has become increasingly popular. Despite the immersive experience for users, its sheer data volume, even with the latest H.266 coding and viewport adaptation, remains a significant challenge to today's networks. Recent studies have shown that integrating deep learning into video coding can significantly enhance compression efficiency. Albeit with increased computational overhead, it brings new opportunities. In this work, we conduct a comprehensive analysis of the potential and issues in applying neural codecs to 360° video streaming. We accordingly present NETA, a synergistic streaming framework that merges neural compression with traditional coding techniques, seamlessly implemented within the edge intelligence. To address the non-trivial challenges in the short viewport prediction window and time-varying viewing directions, we propose implicit-explicit buffer-based prefetching grounded in visual saliency and bitrate adaptation with smart model switching around viewports. A novel Lyapunov-guided deep reinforcement learning algorithm is developed to maximize user experience and ensure long-term system stability. We further discuss the concerns towards practical development and deployment and have built a working prototype that verifies NETA's excellent performance, for instance, a 27% increment in viewing quality, a 90% reduction in rebuffering time, and a 64% decrease in quality variation on average, compared to state-of-the-art approaches.**

***Index Terms*—Neural codec-empowered 360° video, Synergistic streaming, Implicit and explicit buffers, Lyapunov-guided DRL.**

I. Introduction

THE 360° videos with Ultra-High-Definition (UHD) have garnered great attention across various applications, including virtual reality, remote education, and entertainment [1]–[5]. They offer users an immersive visual and acoustic experience through Head-Mounted Displays (HMDs) such as VIVE Cosmos [6] or Google Cardboard paired with a smartphone [7]. However, streaming 360° video poses a significant challenge due to large data volume, requiring extremely high bandwidth, for instance, 200-300Mbps or even higher [1].

While popular tile-based viewport-aware schemes (e.g., [2], [3], [8]–[11]) can mitigate data transfer, the users' Quality of Experience (QoE) still suffers from insufficient end-to-end bandwidth [12], [13]. For example, recent measurements of over 23 million users reveal that the standalone 5G network offers only around 40 Mbps download speed [14]. This dilemma arises primarily from the gap between high-quality requirements and inadequate video compression. Traditional codecs (e.g., H.264 [15], H.265 [16], and H.266 [17]) utilize a series of well-designed, lightweight, handcrafted modules to eliminate data redundancy. Yet, they fall short of end-to-end rate-distortion optimization. Furthermore, they exhibit a poor compression ratio for tiles, especially those within the viewport, primarily due to the loss of spatio-temporal correlation [13], [18]. Our preliminary studies show that the *bits per pixel* (Bpp) of highly salient tiles, which tend to be more attractive to users (i.e., located within viewport) [2], [19], increases by 19% compared to the entire video.

Recently, emerging *neural codecs* leverage deep neural networks (DNNs) to compress video signals as stacked *features* [20]. Through end-to-end rate-distortion trade-off with advanced learning models, they can potentially achieve much higher compression with comparable visual quality as traditional coding [21], [22]. For instance, our content-aware model DVC$^+$ can reduce the Bpp of tiles by 28% over H.266 codec. Yet, pure neural coding-based 360° video streaming remains far from being practical for its ultra-high demand on computing resources and specialized hardware, e.g., GPUs for acceleration. This is particularly challenging for decoders that are often built on lightweight mobile devices of HMDs.

To this end, for the UHD experience, we present a synergistic streaming scheme named NETA, which uniquely combines neural codec with traditional compression techniques, seamlessly implemented within an edge intelligence framework [23], [24]. Specifically, building on the viewport awareness, we utilize the neural codec to compress high-quality tiles to serve the user viewport, thereby improving the visual experience and reducing bandwidth requirement, while low-quality tiles in non-viewport regions, to prevent the blank screens incurred by inaccurate viewport prediction, are encoded using the traditional method to avoid excessive decoding computation and time consumption. At runtime, neural decoding is offloaded to the edge to overcome the lack of end-device resources and achieve hardware-accelerated DNN computations, providing client-friendly services [9], [25], [26].

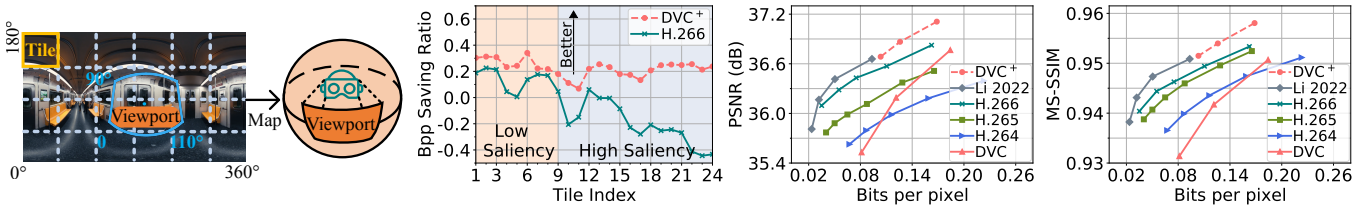
To implement NETA in practice, we also face the following

Manuscript received 18 July 2023; revised 17 March 2024 and 30 June 2024; accepted 18 August 2024. This work was supported in part by the National Natural Science Foundation of China under Grant 62172241. The Associate Editor coordinating the review of this manuscript and approving it for publication was Dr. Zhi Liu. *(Corresponding author: Jiangchuan Liu and Lingjun Pu.)*

Jianxin Shi is with the College of Computer Science, the Key Laboratory of DISSec, the Institute of Systems and Networks, Nankai University, Tianjin 300071, China, and also with the School of Computing Science, Simon Fraser University, Burnaby, BC V5A1S6, Canada. (e-mail: shijianxin@mail.nankai.edu.cn)

Miao Zhang, Linfeng Shen, and Jiangchuan Liu are with the School of Computing Science, Simon Fraser University, Burnaby, BC V5A1S6, Canada (e-mail: mza94@sfu.ca; linfengs@sfu.ca; jcliu@sfu.ca)

Lingjun Pu and Jingdong Xu are with the College of Computer Science, the Key Laboratory of DISSec, the Institute of Systems and Networks, Nankai University, Tianjin 300071, China. (e-mail: pulingjun@nankai.edu.cn; xujd@nankai.edu.cn)



(a) User viewport and video tiling. (b) Compression efficiency. (c) PSNR comparison. (d) MS-SSIM comparison.

Fig. 1: The viewport-aware 360° video representation and performance comparison of different compression techniques.

non-trivial challenges: *(C1) Short viewport prediction window.* While viewport prediction methods are well-studied in accuracy, they typically have a limited prediction window, such as 3 seconds [4], [12], [27]. This necessitates clients to maintain a small buffer size, smaller than the prediction window, to prefetch video chunks over the Internet [2], [8], [13], [28]. However, this often fails to fully utilize the available bandwidth in the face of the various network conditions, resulting in low viewing quality and long rebuffering time [2], [29]. *(C2) Time-varying viewing directions*. In the immersive experience, users can freely move their viewport to focus on different scenes according to their preferences. Due to the inherent geometric distortions introduced by sphere-to-plane projection (e.g., Equirectangular), the number of tiles covered by user viewports in diverse directions varies [30]. Consequently, this results in uncertainties in the transmission requirements and decoding computation burdens for neural chunks. Moreover, the processes of transmission and computation are coupled, further complicating their adaptation scheduling in various network and computation conditions.

As such, to address the first challenge, we present: *(1) Implicit-explicit buffer-based prefetching.* Recent works have shown the effectiveness of a long buffer against network fluctuations [29], [31]. Thus, the visual saliency-aware prefetching with a long buffer is advocated [30], leveraging the degree to which areas attract user attention [19], [32], [33]. However, direct prefetching without regard to the actual user viewport is inadvisable because of the highly redundant transmission and neural computation (i.e., caused by the mismatched user viewport). To this end, we adopt a more nuanced strategy. According to tile saliency, we prefetch the low-quality tiles for broad coverage of future non-viewport regions using an extra *long implicit buffer*. High-quality tiles are prefetched by the playback buffer (i.e., called *short explicit buffer*), directly targeting the user viewport.

To tackle the second challenge, we propose: *(2) Joint bitrate and model adaptation (JBMA).* The neural decoder with larger complexity (e.g., depth) or higher Bpp input (a.k.a encoding bitrate) has a better visual experience but a longer delivery and decoding time, and vice versa [22], [34] (see Fig. 7). Therefore, we develop a joint adaptation scheduling to decide the download bitrate of viewport neural tiles (i.e., compressed by neural methods) and the appropriate decoding model according to current network conditions and computing requirements. After that, those high-quality viewport tiles are filled into the conventional short explicit buffer for display.

Furthermore, an edge node typically serves multiple users. A single user should not be allowed to monopolize all computational resources, thus violating fairness among users. To this end, we introduce a long-term soft constraint to ensure the chunk-average decoding computing cost of the user cannot exceed a predefined budget. The critical JBMA problem is formulated as a constrained integer nonlinear programming with the goal of maximizing user QoE. We develop a novel Lyapunov-guided deep reinforcement learning (DRL) algorithm to optimize the user viewing experience and guarantee long-term system stability.

We implement a prototype of NETA and conduct extensive evaluations and ablation studies over the broadband and 5G network emulations. The results show that NETA improves viewing quality by 27%, reduces rebuffering time by 90%, and decreases quality variation by 64% on average, compared to advanced approaches (i.e., ProbDASH [35], DRL360 [27] and PARSEC [36]). In this work, our main contributions are summarized as follows:

- To our best knowledge, this is the first work to propose a neural compression-empowered, traditional coding-assisted synergistic streaming paradigm for 360° video, seamlessly integrated within edge intelligence framework.
- Our NETA presents novel implicit-explicit buffer-based prefetching grounded in saliency and JBMA scheme, addressing non-trivial challenges in the short viewport prediction and time-varying viewing directions.
- We develop a Lyapunov-guided DRL algorithm to solve the joint adaptation problem of data transmission and neural computation with a long-term fairness constraint.
- We implement a prototype of NETA and demonstrate its excellent performance over other advanced works.

The rest of this work is organized as follows. Section II presents the background and motivation. Section III describes the system designs, formulates the JBMA adaptation problem, and develops the Lyapunov-guided DRL algorithm. Section IV presents the implementation details and evaluates the NETA's performance. Section V discusses key concerns towards practical development and deployment. Section VI introduces the related works. Section VII concludes the paper.

II. Background and Motivation

360° video representation and tiling. 360° videos present an immersive experience by rendering two-dimensional flat views (e.g., Equirectangular map) onto the inner surface of a 3D sphere, as depicted in Fig. 1(a). At any given time, the user located in the center of the sphere can only view a small

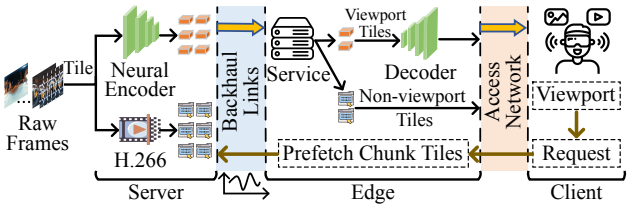
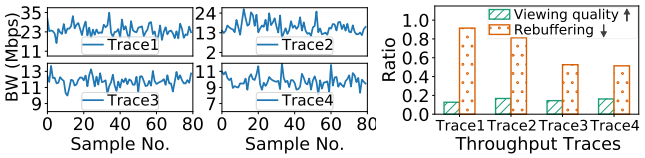


Fig. 2: The workflow of NETA.



(a) Throughput traces. (b) Gain with long buffer.

Fig. 3: Streaming performance under different buffer sizes.

portion of the scene, referred to as a viewport [12], [37]. To optimize 360° video streaming, viewport-aware adaptive bitrate (ABR) schemes have been developed. This involves dividing the panoramic content into tiles, allowing each piece to be encoded, delivered, and rendered independently, thereby improving efficiency and user experience. Following the existing studies [2], [13], we employ a conventional 4 rows × 6 columns tiling strategy in this work. In practice, GPAC player [38] already supports this tile-based streaming.

Limitation: 360° video encoded by traditional codecs. Traditional codecs can support real-time decoding in HMD devices. Still, they suffer from limited performance in redundancy reduction due to individual, handcrafted modules, such as motion estimation, motion compensation, and transform coding, and lack of end-to-end optimization [21], [39]. Even worse, they exhibit lower compression performance for tiles than the entire video due to the loss of spatio-temporal correlation caused by video tiling [13], [18]. Fig. 1(b) illustrates the Bpp saving ratio of tiles as compared to the entire 4K video Jockey [40] with H.266 under the same peak signal-to-noise ratio (PSNR). It can be seen that the Bpp of highly salient tiles with H.266 increases by 19% than the whole video. Those tiles tend to be located within the user's viewport [2], [19]. Consequently, only viewport-aware solutions are insufficient to support UHD 360° video streaming, and user QoE still suffers from inadequate end-to-end bandwidth [14]. Some studies have also proposed super-resolution (SR) based streaming strategies [5], [36], [41], which focus on reconstructing high-resolution video content from a low-resolution input. Our previous work [30] particularly introduced a saliency-aware prefetching scheme to alleviate SR computation and improve the streaming experience. However, their SR process still relies on conventional coding methods, which unfortunately overlook the adverse effects of distortions introduced by conventional codecs on SR-enhanced videos.

Evolution: DNN-based video compression. With automatically learned manners and a single rate-distortion loss function, DNN-based codecs can achieve much higher compression efficiency. From Fig. 1(c) and Fig. 1(d), it can see that Li 2022 [21] already performs better than the latest H.266 codec [17] in terms of PSNR and multi-scale MS-SSIM [42] for the tiled 4K video Jockey [40]. Similar results can also be observed in terms of other metrics [21], [22], [39], such as perceptually based SSIM [43] and LPIPS [44]. While these generic models (e.g., [21]) exhibit superior performance in the public test datasets, they may not be suitable for every video in streaming services. To ensure reliable and superior performance, we advocate a content-aware model for each video based on the existing codec model, leveraging DNN's overfitting property [36], [45]. In this work, we develop the content-aware DVC$^+$ model based on existing DVC [39]. As in Fig. 1(b), the DVC$^+$ can reduce the Bpp of tiles by 28% as by H.266 codec. It also maintains a high compression performance under different Quantization Parameters (QP) as shown in Fig. 1(c) and Fig. 1(d).[1] However, pure neural coding-based 360° video streaming remains far from being practical, as it requires ultra-high computing resources for neural decoding [21], [22]. This is particularly challenging for lightweight HMDs.

Insight. To this end, we advocate the development of a synergistic solution. Leveraging the property of viewport awareness, we respectively exploit neural methods to compress the high-quality tiles for the viewport region and traditional codecs to encode low-quality tiles for the non-viewport region to improve the viewing experience, adapt inadequate bandwidth, and reduce computing.

III. System Mechanism

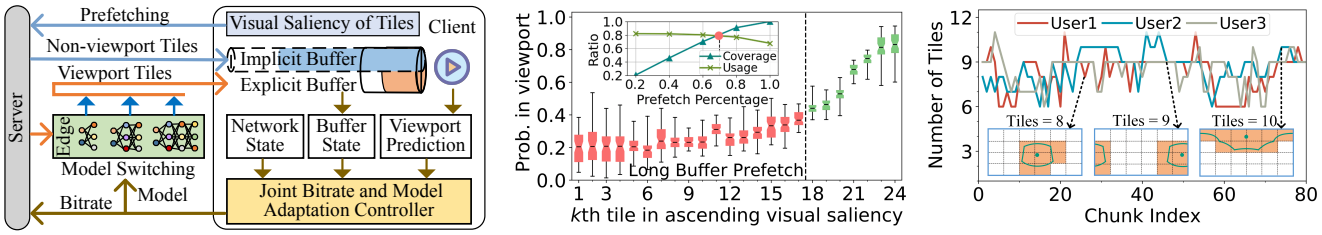
In this section, we present the designs of NETA, including the basic components, well-suited designs, and interactions. Then, we formulate the critical bitrate and model adaptation problem and develop a Lyapunov-guided DRL algorithm.

A. NETA Overview

The hybrid video compression scheme can significantly reduce DNN computing workload, but hardware acceleration is critical for the success of such neural codec-empowered 360° video streaming. Generally, HMD devices are lightweight and have insufficient computing resources [6], [7]. This means scheduling excessive workloads on the headset can easily run into heat dissipation issues and computational time bottlenecks, resulting in a poor viewing experience [46]. Therefore, we advocate for edge intelligence services to offload the neural decoding operations for end-device power and computing savings, as depicted in Fig. 2.

Server side. The server applies the conventional Dynamic Adaptive Streaming over HTTP (DASH) protocol to deliver video data [47], where each video is divided into serialized chunk tiles. The difference is that high-quality copies (i.e., video streams encoded with different bitrates) are compressed

[1]Note that the DVC$^+$ can be substituted by other superior models, which can serve as a black-box without discussing the internal codec details and logic. Also, we focus only on the inference time of the neural network, while the time costs executed in the CPU of others, such as arithmetic coding, can be concurrent with the GPU process and are already well optimized in conventional codecs.



(a) Well-suited designs of NETA. (b) Relationship between saliency and viewports. (c) Number of tiles covered by viewport.

Fig. 4: System mechanism for short viewport prediction and time-varying viewing directions in NETA.

by the neural codec (e.g., DVC$^+$), and the low-quality copies are encoded by the traditional method (e.g., H.266). The neural codec consists of the encoder and decoder, which are trained together and deployed offline. Thanks to neural codec's superior compression efficiency, high-quality copies possess a smaller data volume than traditionally encoded copies. As a result, NETA can help achieve higher efficiency in terms of content caching. At runtime, once the server side receives bitrate decision results, it immediately delivers the corresponding tile content. Meanwhile, all video information, such as video duration, tile size, content quality, and tile saliency value, is encoded in the Media Presentation Description (MPD) file, which is only dozens of KB in size, and available to the client as an HTTP resource.

Client side. The client first predicts the future user viewport, such as by analyzing the viewing trajectory, then generates the bitrate requests for the chunk tiles and sends them to the edge node. Simultaneously, the client receives requested tiles from the edge and subsequently renders them to the user.

Edge side. Edge intelligence is widely popular in emerging networks because of its ability to bring powerful DNN computation close to the users, as illustrated in Fig. 2. The edge is typically deployed with a network access point, such as the router, WiFi, or micro base station [9], [23], [48]. In NETA, the edge processes tile requests from the client and forwards them to the server. Upon receiving the viewport tile from the server, the edge decodes it with the neural decoder and delivers it to the client, while the non-viewport tile is transmitted directly to the client. It is worth noting that the delivery delay over the access network can be negligible due to the decoding and transmitting concurrency at the tile or frame level and sufficient access bandwidth (e.g., up to 1000Mbps in the 5G cellular and WiFi 6, as measured by [49]). This is also confirmed by recent work [50]. The pre-trained decoder model is cached at the edge in advance and can be updated if necessary or during network idle periods. The updated policy is flexible and can draw on existing schemes [51], [52]. Notably, NETA does not require any additional information from the client compared to regular solutions [2], [3], which avoids potential privacy leakage.

Summary. (i) NETA utilities neural codec to alleviate the bandwidth burden of backhaul links and improve the viewing quality; (ii) NETA offloads neural decoding to the edge, providing client-friendly services; (iii) NETA incorporates viewport awareness to reduce computing requirements and prevent bandwidth waste.

B. Well-suited Designs

Despite the significant advantages of NETA, it is required to address the following non-trivial issues:

Q1: *How to mitigate the adverse effects of the short viewport prediction window?* The UHD 360° video streaming typically involves viewport-aware schemes to mitigate data usage. Yet, recent studies [3], [28] have indicated that accurate viewport prediction can only be reliably achieved within a short window (i.e., 3s). Since prefetched video content duration cannot exceed the viewport prediction window, streaming schemes are compelled to deploy a small playback buffer to prefetch video content over the network [2], [13], [53]. This, however, often proves inadequate to handle bandwidth fluctuations and fails to fully utilize available network throughput [29], [30]. For clarity, similar to our previous work [30], we evaluate the user experience under a long buffer of 30s and a small buffer of 3s. The downstream bandwidths are emulated with four real throughput traces (see Fig. 3(a)), which are randomly chosen from the latest FCC dataset [54]. The detailed settings are in §IV. As shown in Fig. 3(b), utilizing a large buffer achieves a 15% increment in viewing quality and a 69% reduction in rebuffering time compared to the small buffer under different traces.

Response 1: *Implicit-explicit buffer-based prefetching.* Likewise, we advocate a long buffer size. Here, we use visual saliency as prior information to prefetch only low-quality traditional tiles for later non-viewport regions far in advance.[2] This is to avoid potentially redundant transmissions and neural computations due to viewport mismatches as much as possible. As in Fig. 4(a), the long prefetch buffer is known as *Implicit Buffer* (LIBuf), and the original playback buffer is called the *Short Explicit Buffer* (SEBuf). Prefetched non-viewport tiles are cached in LIBuf and rendered with viewport tiles in SEBuf together. Fig. 4(b) illustrates the relationship between the tile visual saliency (generated by [55]) and user viewports (viewing records of 48 users) for 360° video Help [56]. It indicates that tiles with lower saliency are more likely to be in non-viewport regions. Prefetching more low-saliency tiles gives better coverage of non-viewport regions. Still, it reduces tile usage (i.e., defined as the ratio of used tiles in future non-viewport regions to total prefetch tiles), as depicted in the subfigure of Fig. 4(b). When 70% of low-saliency tiles

[2]Note that non-viewport tiles with low-quality are indispensable, as they help prevent poor user experience resulting from inaccurate viewport prediction. These tiles cover about 70% of the scenes and require a large amount of data transfer, as shown in Fig. 4(c).

are prefetched, there is a natural trade-off between coverage and tile usage (i.e., red dot). Based on this, we strategically prefetch low-quality tiles with saliency in the bottom 70% for each chunk using LIBuf to cover areas of lower user interest. Despite its simplicity, this prefetching scheme has been shown to perform well in our system. Note that our previous Sophon [30] prefetches both low and high-quality tiles based on their saliency with the unified long buffer, without tailoring to the user viewport. In contrast, NETA adopts a more nuanced strategy by prefetching low-quality tiles using LIBuf for broader coverage, and high-quality neural tiles using SEBuf for better viewing quality.

Q2: *How to cope with uncertain transmission and computing requirements arising from time-varying viewing directions?* 360° videos offer a wealth of content, allowing users to explore different scenes based on their preferences by moving the viewport. Nevertheless, due to projection mapping, the number of tiles within the user's viewport varies significantly across serialized chunks. Fig. 4(c) illustrates the variation in the number of viewport tiles as the chunk index increases for three users randomly selected from 48 users [56]. This results in uncertain data transmission and decoding calculations among chunks. It is highly challenging for on-time delivery and stable playback. This situation is further exacerbated by the constant changes in available backhaul bandwidth [34] and edge computing resources [23].

Response 2: *Joint bitrate and model adaptation (JBMA).* The data transfer and neural computing are both the time bottleneck and easy to trigger the rebuffering. To this end, NETA must adapt to the various situations in both transmission and computation. The inference time of the model is related to its complexity [22], [34] (see Fig. 7). Thus, different from the traditional ABR, we adaptively make the joint decisions for the bitrate of viewport tiles and complexity of the decoder model according to states of the network, computing, and buffer, as depicted in Fig. 4(a). Although our previous work [30] also faces the network adaptation and model selection-based computation adaptation, it treated them in isolation, overlooking the crucial interplay between them. In contrast, we consider the consecutive and coupled network and computation process as a joint optimization task in NETA. Here, we advocate that the JBMA controller remains deployed on the client rather than the edge, as in other neural-enhanced solutions [34], [50], for user privacy security and compatibility with current DASH. In addition, it is important to prevent a single user from monopolizing edge computing resources without any limits.

System goal. Our objective is to select the appropriate input bitrate and decoding model to maximize user QoE under various situations while maintaining fairness among the users.

C. Problem Formulation

Video information. As shown in Fig. 2, the serialized chunks with fixed-duration (e.g., $\delta = 1$ second) are denoted as $\mathcal{C} = \{C_1, C_2, ..., C_I\}$. Each tile is compressed into multiple high-quality levels by the neural encoder, indicated as $\mathcal{L} = \{L_1, L_2, ..., L_K\}$, which are served for viewport regions. According to the viewport prediction for the chunk C_i, the tiles within the user viewport are indexed by $J_i = \{V_{i1}, V_{i2}, ..., V_{ij}, ...\}$. At the same time, a low-quality copy encoded by the traditional codec (e.g., H.266) is served for non-viewport regions. For chunk C_i, to correct the incomplete prefetching of the visual-saliency-based scheme, it is required to download those non-viewport tiles that are not being prefetched (i.e, not in *LIBuf*) in advance, and their total size is denoted as D_i^l. Note that downloading the current chunk tiles has a higher priority than the visual-saliency-based prefetching to avoid competition for bandwidth resources. The optional decoding models in the edge are indexed by $\mathcal{N} = \{N_1, N_2, ..., N_M\}$.

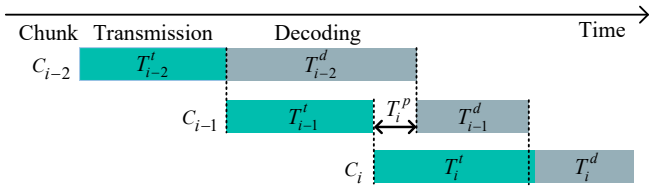


Fig. 5: Transmission and neural decoding processes.

Control variables. Unlike the [30], NETA necessitates the simultaneous decision-making for the downloaded bitrate and the decoding model due to their interdependence. The bitrate decision is denoted by $x_i(k), \forall k \in \{1, 2, ..., K\}$, where $x_i(k)$ signifies that the client requests the quality L_k for viewport tiles in chunk C_i. The size of the tile V_{ij} at quality L_k is denoted by $D_{ij}(k)$. For model switching, we use $y_i(m), \forall m \in \{1, 2, ..., M\}$, where $y_i(m)$ indicates that model N_m is used for neural decoding.

Objective function. Typically, the users' QoE is captured using three metrics. Still, their formulations in NETA are significantly different from the previous works [27], [28], [36] due to the effects of the decoding computation process: *(i) Viewing quality* (Q_i). Since neural decoding performance varies across models and input Bpp levels, following recent works (e.g., [10], [12], [28], [35], [57]), we adopt the PSNR (included in MPD file) instead of the traditional bitrate as viewing quality, denoted as $f_j(x_i(k), y_i(m))$, which provides a straightforward indication of the fidelity of the reconstructed tile compared to the original. Notably, the viewing quality considered here does not depend on the specific metric, and other metrics like SSIM, MS-SSIM, and LPIPS, can also be widely used [2], [37], [46]. Thus, the average viewport quality of chunk C_i is formulated as $Q_i = \frac{1}{|J_i|} \sum_{j=1}^{|J_i|} f_j(x_i(k), y_i(m))$. *(ii) Rebuffering time* (R_i) that represents how long the video stalls because the *SEBuf* is empty. Note that, different from traditional streaming, the rebuffering time of chunk C_i in NETA is incurred by both the transmission latency (i.e., $T_i^t = (\sum_{j=1}^{|J_i|} D_{ij}(k) + D_i^l)/BW$, where BW indicates currently available bandwidth) and neural decoding computing time (i.e., $T_i^d = \sum_{j=1}^{|J_i|} g_j(x_i(k), y_i(m))$, where $g_j(x_i(k), y_i(m))$ denotes the inference time for tile V_{ij}). As in Fig. 5, the transmission of the current chunk C_i and the decoding process of the last chunk C_{i-1} are carried out in parallel. Hence, the rebuffering time is formulated as $R_i = [(T_i^t - T_{i-1}^d - T_i^p)^+ + T_i^d - B_i]^+$, where

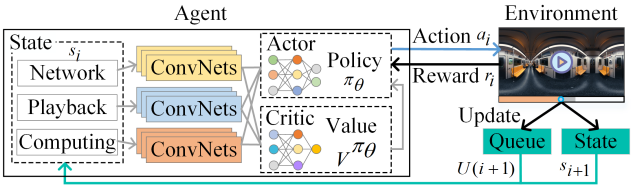


Fig. 6: Illustration of the proposed Lyapunov-guided DRL.

$(.)^+ = \max\{.,0\}$; $T_i^p = (T_{i-2}^d - T_{i-1}^t + T_{i-1}^p)^+$ indicates the available download time inherited from the previous chunks; and $B_{i+1} = [B_i - T_i^d - (T_i^t - T_{i-1}^d - T_i^p)^+]^+ + \delta$ denotes *SEBuf* occupation. *(iii) Smoothness* (S_i) that measures quality variation between two consecutive chunks: $S_i = |Q_i - Q_{i-1}|$. Formally, the objective of the system is to maximize user QoE, represented as follows:

$$\max \frac{1}{I}\sum_{i=1}^{I}(Q_i - \alpha R_i - \beta S_i), \tag{1}$$

where α and β are adjustment factors to control the penalties of rebuffering and smoothness [13], [28], [36]. Although the occasional concurrency of transmission and computing for tiles in the chunk can further bring benefits, we do not consider it in this work.

Constraint. To guarantee fairness among multiple users sharing the same edge node, we introduce a long-term soft constraint that restricts the chunk-average computing cost of a single user from exceeding a predefined budget threshold W:

$$\frac{1}{I}\sum_{i=1}^{I}\sum_{j=1}^{|J_i|} h_j(x_i(k), y_i(m)) \le W. \tag{2}$$

where $h_j(x_i(k), y_i(m))$ denotes the computing cost of tile V_{ij}.

Analysis. The network and computation processes in JBMA (1) are sequential and coupled. This allows NETA to dynamically adjust both the download quality and decoding model in response to changing network conditions and computational requirements, ensuring an optimized streaming experience that balances quality and resource utilization efficiently. Yet, the online JBMA is characterized as an integer nonlinear programming, marking it as a nonconvex problem. Given the time-varying bandwidth and viewing directions, ensuring long-term fairness when making decisions for each chunk becomes more challenging without *priori* knowledge.

D. Lyapunov-guided DRL

Insight. Considering the Markov property inherent in formula (1), Deep Reinforcement Learning (DRL) [27], [30], [58] emerges as a promising solution for handling this sequential decision problem, especially in dynamic environments. For instance, our previous Sophon [30] employs a DRL to navigate the SR model-aware computing adaptation. However, despite its efficiency, DRL cannot directly tackle the constraint of long-term computing stability (2). This highlights the necessity for integrating additional mechanisms or strategies within the DRL framework. Fortunately, the Lyapunov method [29], [59] can be used to trade off utility maximization and stability control by decoupling multi-stage stochastic optimization into sequential per-stage deterministic subproblems, while providing the theoretical guarantee of long-term system stability. As a result, we develop a lightweight Lyapunov-guided DRL (LyaDRL) that can make near-optimal decisions in real-time. To the best of our knowledge, this is the first time that the Lyapunov-guided DRL framework is used in video streaming. Previous researchers mainly employ it for resource scheduling and orchestration [59], [60].

Lyapunov-based decoupling. We first resort to the Lyapunov drift-plus-penalty method [59] to decouple the JBMA problem into per-chunk subproblems. To cope with the fairness constraint in (2), we introduce a virtual queue $U(i)$ to keep track of the computing cost. Specifically, we set $U(0) = 0$ and update the queue as:

$$U(i) = [U(i-1) + H(x_{i-1}(k), y_{i-1}(m)) - W]^+, \tag{3}$$

where $H(x_{i-1}(k), y_{i-1}(m)) = \sum_{j=1}^{|J_{i-1}|} h_j(x_{i-1}(k), y_{i-1}(m))$ denotes the total computing cost of chunk C_{i-1}. After that, the original JBMA problem can be transformed into the following per-chunk optimization subproblem:

$$\max \mathbb{E}[V\cdot(Q_i - \alpha R_i - \beta S_i) - U(i)(H(x_i(k), y_i(m)) - W)]. \tag{4}$$

where V is a predefined positive parameter that is widely used in the Lyapunov drift-plus-penalty method [29], [59]. With the increment of V, the JBMA becomes more progressive in bitrate and model selection to obtain a better user experience.

Theorem. The chunk-average queue size $U(i)$ is strongly stable with $\lim_{I\to\infty} \frac{1}{I}\sum_i \mathbb{E}[U(i)] = \mathcal{O}(V)$. Meanwhile, the average penalty is upper-bounded as $r^* - \lim_{I\to\infty} \frac{1}{I}\sum_i \mathbb{E}[Q_i - \alpha R_i - \beta S_i] = \mathcal{O}(\frac{1}{V})$, where r^* is the optimal QoE value. The detailed derivation please refers to the works [59], [60].

DRL for per-chunk optimization. The lifetime of the chunk-based JBMA (4) can be regarded as a Markov Decision Process (MDP) task. We drive a DRL solution through the interaction between *agent* (i.e., the decision-maker) and *environment*, as shown in Fig. 6. For each chunk, the agent picks an *action* based on the current environment *state*, using a trained *policy*. And then, the environment executes the action and feeds back a *reward*. Next, the agent acquires a new state and takes action for the next chunk.

For chunk C_i, *(i) State* (s_i) contains the network, playback, and computing information: $s_i = (\vec{E}_i, \vec{N}_i, \vec{T}_i, \vec{O}_i, \vec{D}_i, U_i, P_i, B_i, Q_{i-1})$, where $\vec{E}_i$, $\vec{N}_i$, $\vec{T}_i$ and $\vec{O}_i$ present the actual network throughput, neural processing time acquired from the edge, delivery time and number of viewport tiles (i.e., streaming and computing burden) for the past $n = 8$ chunks respectively; $\vec{D}_i$ indicates total download size under different bitrate levels; and P_i denotes number of viewport tiles. *(ii) Action* (a_i). The action space consists of the neural bitrate set $\mathcal{L}$ and the decoding model set $\mathcal{N}$. Decision $a_i = (x_i(k), y_i(m))$ indicate that the client requests bitrate L_k and uses model N_m for decoding. *(iii) Reward* (r_i). We take the objective (4) as the reward. This is different from the previous DRL-based solution [27], [30], [58], where they directly employ the user QoE as the reward. The agent aims to learn the policy $\boldsymbol{\pi}$ from the state to maximize the cumulative reward $R = \sum_{i=1}^{I} \gamma^i r_i$, where γ is the discounting factor. We exploit an actor-critic framework [27], [48] for on-policy training.

Actor module. The decision policy $\boldsymbol{\pi}_\theta(s_i, a_i)$ is represented as an *actor* network, where its parameter θ is updated by a policy gradient method: $\theta \leftarrow \theta + \eta \sum_i \bigtriangledown_\theta log\boldsymbol{\pi}_\theta(s_i, a_i) A^{\pi_\theta}(s_i, a_i)$, where η is the learning rate. $A^{\pi_\theta}(s_i, a_i)$ is the advantage function, defining the difference between the expected total reward when the agent deterministically picks a_i in s_i and expected reward for actions based on policy $\boldsymbol{\pi}_\theta$. The actor module passes vector states to identical 1D convolutional layers (ConvNets) with 128 filters. The constant states are passed into fully connected layers (FC) with 128 neurons. The outputs from the previous layers are then aggregated into a hidden layer with 128 neurons. Finally, the Softmax function is adopted, and its output corresponds to the action space.

Critic module. To calculate the advantage function for a given experience, we require an estimate for the expected cumulative reward starting at state s_i and following the policy π_θ, which is denoted as the value function $V^{\boldsymbol{\pi}_\theta}(s_i)$. A critic network with parameters ω is used to learn the estimate of value function from previously acquired rewards so as to assist in training the policy. The parameter ω of the critic network is updated as $\omega \leftarrow \omega - \eta \sum_i \bigtriangledown_\omega (r_i + \gamma V^{\boldsymbol{\pi}_\theta}(s_{i+1}; \omega) - V^{\boldsymbol{\pi}_\theta}(s_i; \omega))^2$. For an experience (s_i, a_i, r_i, s_{i+1}), the advantage is calculated as $A^{\boldsymbol{\pi}_\theta}(s_i, a_i) = r_i + \gamma V^{\boldsymbol{\pi}_\theta}(s_{i+1}; \omega) - V^{\boldsymbol{\pi}_\theta}(s_i; \omega)$. The critic adopts the same network structure as the actor, but its final output is a linear neuron. For each filter of convolutional layers, the size is 4, and the stride is 1. All fully connected layers use the activation function ReLU. The learning rate η is 10^{-4}, and discounting factor γ is 0.99. We train the DRL model offline with a simulation program, followed by online fine-tuning and deployment [27], [48].

Compared with the traditional DRL, the proposed Lyapunov-guided DRL first adopts the drift-plus-penalty method [59] and introduces a virtual queue to decouple the original JBMA problem. Then, by periodically interacting with the playback environment, DRL is used to train a decision strategy and select appropriate actions for the sequential subproblems based on the states and size of the virtual queue. As a result, it can maintain queue stability, thereby meeting the long-term computing constraint (formula (2)). Meanwhile, it provides a theoretical performance guarantee.

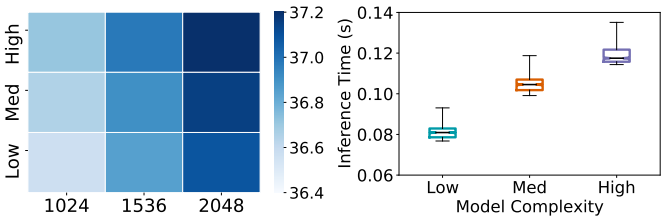
IV. Prototype Evaluation

Here, we develop a prototype of NETA and conduct extensive experiments by answering the following questions.

- **Q1:** How does the overall performance of NETA compare to other state-of-the-art solutions?
- **Q2:** How does each design component of NETA contribute to the overall performance?
- **Q3:** Can NETA be effectively applied to various types of videos and time-varying viewing directions?
- **Q4:** What about the overhead involved in NETA?

A. Implementation and Settings

Testbed. Our test platform comprises three desktop computers, serving as the server, edge, and client, respectively. All desktops are connected to a gigabit router. The server,



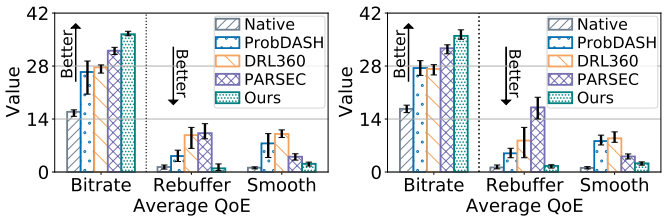
(a) PSNR comparison. (b) Decoding time for tile.

Fig. 7: The performance of DVC$^+$ with different complexity.

operating on Ubuntu 20.04 LTS, employs Apache Tomcat software to provide DASH streaming services. The edge, outfitted with an NVIDIA GeForce RTX 3090 GPU, executes a proxy script developed in Python to facilitate the neural decoder and data forwarding. The client utilizes the open-source GPAC player [38] to assess user experience. The client maintains a 30s (recommended in [29]) implicit long buffer for visual-saliency-based prefetching and a 3s explicit buffer (consistent with [27], [36]) for video playback.

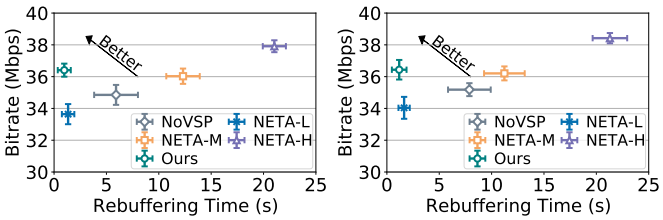
Network emulation. The influence of network conditions, such as delay, jitter, and packet loss, is abstracted as end-to-end throughput. Consequently, we employ two real-world throughput datasets: *(i) FCC* [54], the American fixed broadband measurements (i.e., 2022.06-2022.12); *(ii) 5G* [61], a 5G dataset collected from a major Irish mobile operator (utilizing traces in the static pattern). In line with [27], we randomly select some throughput traces averaging 20Mbps, corresponding to the 60th and 40th percentile values of CDF in FCC and 5G, respectively. All traces are emulated using the server's built-in Linux Traffic Control tool.

Videos and visual saliency. We utilize three distinct categories of 360° videos and their corresponding viewing trajectories gathered from 48 subjects [56]: *Help* (Film), *Surfing* (Sport), and *Rhinos* (Scenery). We employ PARIMA [3] to predict the viewport for the subsequent 3s. The low-quality tiles in the non-viewport region are prefetched to prevent blank screens caused by inaccurate prediction. In addition, despite the 3s prediction window, the predicted viewport in our system is dynamically updated after each chunk playback to increase its accuracy. To further improve the robustness of system, similar to recent studies (e.g., [13], [46]), we can also deliver a portion larger than the user viewport to tolerate inaccurate predictions and ensure a highly immersive experience. Due to the unavailability of public raw 360° video data, we choose three 4K raw videos (i.e., HoneyBee, Jockey, and FlowerPan) with analogous categories from the UVG dataset [40] to replace the original 360° video content for training and evaluation of neural codec. The raw videos are divided into tiles using FFmpeg [62]. To support benchmark schemes utilizing traditional codecs and bitrate-based adaptation methods, each raw tile is encoded into five quality levels (QP=22,23,24,25,26) using the H.266 codec [17]. The selection of the QP parameters is primarily based on the post-encoding bitrate and the simulated network conditions. Other quality ladders, such as those used in Common Test Conditions [17], can also be employed for evaluation due to the adaptivity of those benchmark schemes.



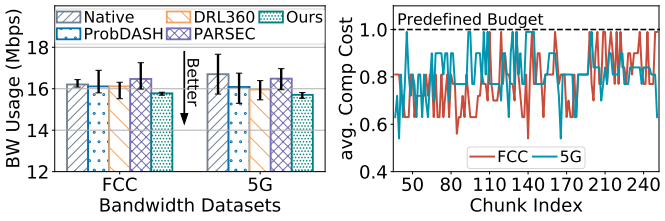
(a) FCC dataset. (b) 5G dataset.

Fig. 8: Performance comparison with SOTA strategies.



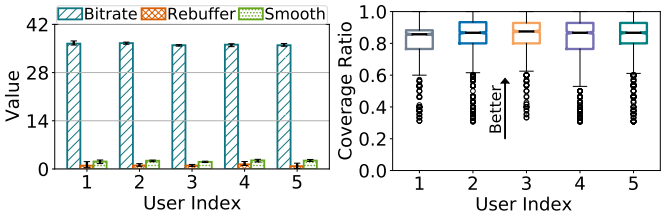
(a) FCC dataset. (b) 5G dataset.

Fig. 10: Performance comparison of ablation studies.



(a) Bandwidth usage. (b) Chunk-average computing cost.

Fig. 9: Service performance of NETA versus other schemes.



(a) QoE of different users. (b) Non-viewport coverage.

Fig. 11: System performance under dynamic preferences.

Note that only the lowest quality level (QP=26) is selected to serve the non-viewport region in NETA. We follow the saliency map acquisition technique proposed by [55], where saliency maps are generated at the frame level. To calculate the tile saliency value, we divide these frame-level saliency maps into tiles. Subsequently, these tiles are grouped together to get tile chunks, based on the chunk duration. The tile saliency value is calculated as the average saliency across all tile maps contained within a chunk. Here, this hybrid dataset is valid for the performance evaluation of NETA since the saliency-based prefetching scheme is solely related to viewing trajectories and saliency maps.

Neural codec. We present a content-aware neural codec called DVC$^+$, developed on top of the publicly available DVC [39]. It leverages the overfitting property of DNNs [36], [45], optimizing the compression process for each video to significantly improve efficiency. Each DVC$^+$ model consists of an encoder and a decoder. Similar to the multiple quality levels of conventional codecs, we generate the three quality copies by adjusting the Lagrange multiplier λ of DVC$^+$. λ is an adjustment factor that determines the trade-off between the Bpp and distortion. Here, the λ is set to '1024', '1536', and '2048', respectively. To handle the variation in computing requirements, we develop the three complexity levels of decoders: 'Low', 'Medium', and 'High' by changing the network depth. Fig. 7 shows the performance and tile inference time under different complexity levels of decoders for video Jockey. The tile computing cost under different complexity decoders is set to 0.07, 0.09, and 0.12, respectively, according to their network depth. The predefined budget threshold W of the chunk and positive parameter V are empirically set to 1 and 10. Note that content providers can easily adjust those parameters based on their service patterns.

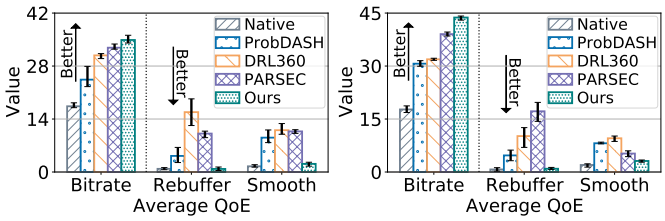
Metrics. *(i) Viewport viewing quality.* For a fair comparison, same as [34] and [36], we map the PSNR quality in NETA back to the playback bitrate (Mbps) of the entire video encoded with H.266: $\text{Bitrate}(i) = \text{PSNR}^{-1}(Q(i))$, where PSNR^{-1} is created based on the coding quality of copies with different QP parameters and piece-wise linear interpolation method. For SOTA benchmarks, the bitrate is directly used to make the evaluation. *(ii) Rebuffering time*, the accumulated total rebuffering duration (s) in video playback. *(iii) Smoothness penalty*, the chunk-average quality variation (Mbps). According to [48] and [27], the adjustment factors α and β are set to 4 and 1 to trade-off the benefits of viewing quality and penalties of rebuffering and smoothness.

Benchmarks. (i) Native, a standard long buffer-based DASH approach without viewport awareness, which employs the BOLA [29] algorithm to assign the maximum bitrate based on the predicted bandwidth. **(ii) ProbDASH** [35], which introduces a target-buffer-based rate control adaptation scheme to prevent the playback rebuffering in viewport-aware bitrate selection; **(iii) DRL360** [27], which applies a DRL method to decide the bitrate of tiles inside the user viewport; **(iv) PARSEC** [36], which explores the available network and the chunk-based micro super-resolution models to maximize the viewing quality of user viewport.

Ablation strategies. To evaluate the effects of visual-saliency-based prefetching and JBMA scheme, we consider four strategies: **(i) NoVSP** only involves the bitrate and model adaptation without the Visual Saliency-based long buffer Prefetching; **(ii) NETA-L**, **(iii) NETA-M**, and **(iv) NETA-H** represent the proposed NETA equipped with fixed Low decoder, Medium decoder, and High neural respectively.

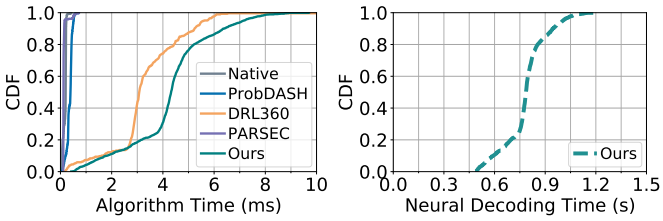
B. Experiment and Analysis

Overall performance (Q1). Fig. 8 shows the performance comparison for video Help. Compared with Native, NETA improves the viewing quality by 1.28× in FCC and by 1.15× in 5G. The main reason is that the Native scheme transmits



(a) Video Surfing (Sport). (b) Video Rhinos (Scenery).

Fig. 12: Performance comparison under different videos.



(a) Algorithm execution time. (b) Neural decoding time per chunk.

Fig. 13: The time cost involved in NETA.

a large amount of non-viewport data with the same quality as the viewport, which wastes limited bandwidth and leads to low viewing quality. Thanks to the long buffer, both Native and our NETA exhibit lower rebuffering time and quality variation. For example, the rebuffering time in NETA is less than 1 second. Compared with ProbDASH and DRL360, NETA achieves a 35% increment in viewing quality in FCC and a 32% increment in 5G, along with a 54% reduction in quality variation in FCC, a 74% reduction in 5G. This is because our solution takes advantage of neural codecs, which offer a higher compression ratio and require less data delivery. However, traditional coding-based schemes often must perform frequent quality variations to adapt to fluctuating bandwidth and guarantee viewing quality. Meanwhile, our solution reduces rebuffering time by 72% compared to ProbDASH and 92% compared to DRL360. This indicates that the short buffer size fails to counter the time-varying network conditions. Although PARSEC utilizes the super-resolution technique to reduce data transfer overhead and enhance user experience, NETA still achieves a 12% improvement in viewing quality and a 46% reduction in quality variation. The reasons are two-fold. First, PARSEC overlooks the fact that distortions incurred by conventional codecs are fed back into the SR-reconstructed video. Second, employing exceptionally low-resolution video as input for the SR process adversely affects the quality of the reconstructed video. Similarly, the small buffer size results in a long rebuffering time for PARSEC. Despite this, NETA still exhibits lower bandwidth consumption, achieving a 4% reduction compared to other schemes, as shown in Fig. 9(a). Meanwhile, Fig. 9(b) confirms the proposed LyaDRL algorithm can ensure that the chunk-average computing cost does not exceed the predefined budget.

Ablation studies (Q2). To better understand performance gains, we evaluate the effects of each design component of NETA. From Fig. 10, it can be seen that the viewing quality of NoVSP decreases by 4% compared to NETA; the rebuffering

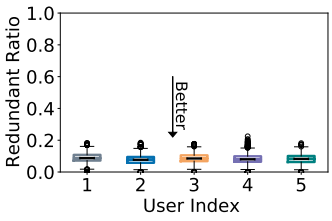


Fig. 14: The redundant transmission involved in NETA.

time of NoVSP experiences a rapid increase, reaching up to 8s on average. This is because NoVSP lacks saliency-based long buffer prefetching and must download both viewport and non-viewport tiles within a short viewport prediction window. As a result, NoVSP has to lower the bitrate level to cope with fluctuating bandwidth. Compared with NETA-L, our proposed NETA achieves an 8% increment in viewing quality in FCC and a 7% increment in 5G. NETA-L, which considers an adaptive bitrate scheme and simply adopts the decoder with the lowest complexity, addresses the issue of time-varying viewing directions by sacrificing viewing quality. Although NETA-M and NETA-H achieve better viewing quality through the more complex decoder models, they incur unacceptable rebuffering time due to the long inference time. The rebuffering time of NETA-M and NETA-H is 12s and 21s, respectively. Meanwhile, they ignore the long-term stability constraint.

Universality of NETA (Q3). Thanks to joint bitrate and model adaptation, our scheme can fight against changing network environments and time-varying viewing directions. Fig. 11(a) presents users' QoE under five real viewing trajectories. It can be seen that the average viewing quality of different users varies by at most 1.2%, and all users experience low rebuffering time and quality variation. As shown in Fig. 11(b), the prefetched low-quality tiles cover 84% actual non-viewport regions on average. This indicates that implicit-explicit buffer-based prefetching can utilize the bandwidth outside the prediction window and help later viewport-aware JBMA reduce data transfer overhead and improve user viewing quality. Furthermore, we evaluated the performance of NETA across different video categories, as shown in Fig. 12. Compared to other advanced schemes, NETA improves viewing quality by 26%, reduces rebuffering time by 89%, and decreases quality variation by 70% on average for the sports video Surfing and the scenery video Rhinos. This indicates the content-aware neural codec can efficiently guarantee reliable and superior compression performance for different videos.

Overhead (Q4). To evaluate the complexity of various schemes, we further measured the algorithm execution time (i.e., running on Intel Core i5-12600K CPU), as depicted in Fig. 13(a). The Native, ProbDash, and PARSEC are notably swift, all under 1ms owing to their lightweight algorithm designs. In comparison, DRL360's execution time stands at 3.4ms on average. Although our LyaDRL algorithm shows a higher average execution time of 4.3ms, it remains considerably lightweight due to chunk-level decision-making, i.e. once per second. In Fig. 13(b), the average decoding time per chunk is 0.88 seconds, which is lower than the chunk duration,

thereby enabling real-time video playback (i.e., 24 frames per second). With the rapid development of acceleration hardware and neural models, the computational burden of neural coding will be gradually alleviated. Due to time-varying viewing directions, the prefetching scheme with a long buffer may download some tiles not used in future real non-viewport regions. Fig. 14 shows there are 6.8% redundant transmissions. Despite this, the bandwidth usage of NETA remains lower than other solutions, as shown in Fig. 9(a). Moreover, the prefetching scheme yields significant performance gains, as evidenced by the ablation studies.

V. Practical Discussion

Training content-aware neural codecs. The content-aware model can counteract the performance variability across diverse videos. Nevertheless, the total computation cost for the content provider may be high when training a separate model for each video from scratch. To address this, the popular technique of fine-tuning [45], [48], derived from the transfer learning approach, can be employed. This ensures superior performance on a specific video while significantly reducing training costs, as demonstrated in Yeo *et al*. [31] (where training costs are reduced by 5-6×). Specifically, a generic model is initially pre-trained on standard datasets. Subsequently, the content-aware model for each video can be generated by fine-tuning the generic model [2].

Fine-grained DNN services in the edge. In NETA, each user will be allocated fixed and dedicated edge computing resources. Content providers perform coarse-grained scheduling to ensure long-term computing stability. Given the benefits of serverless computing in autoscaling resources and pay-as-you-go pricing strategy, fine-grained machine learning services have been explored for video processing tasks (e.g., video analytics [24] and learning-based streaming [9]) in various ways. In light of this, we consider that serverless computing can be integrated into edge nodes to enable fine-grained resource scheduling and management for the multi-user scenario. This poses great challenges that we plan to address in the future.

Practical implementation. With the requirements for lightweight HMD devices, the UHD 360° video application services (e.g., SteamVR [25] and Oculus [26] typically necessitate additional auxiliary computations. These can be provided by the emerging edge intelligence, as evidenced by recent studies [9], [23], [48]. Consequently, the proposed NETA streaming framework can be seamlessly integrated into these existing architectures, enhancing their performance and capabilities. Deploying neural operations within NETA at edge nodes significantly contributes to the conservation of power and computational resources on the end HMD devices. Moreover, the adoption of neural codecs alleviates the bandwidth constraints and enables a high-fidelity viewing experience.

VI. Related Work

Tile-based 360° video streaming. Previous works, such as DRL360 [27], PARIMA [3], Pano [10], Flare [11] and MANSY [63], mainly focused on tile-based viewport adaptation determining tile bitrates given viewport information. STC [64] proposed delivering ShiftTiles along viewport trajectories instead of planar video. Hu *et al*. [65] presented an asymmetrical scheme by adopting the probabilistic viewport adaptation and buffer replacement strategy. PARSEC [36] and FOCAS [41] further investigated the use of super-resolution (SR) to enhance user experience. Ebublio [9] considered an intelligent edge caching framework to optimize multi-user QoE. Despite their effort, those methods overlook the issue in short viewport prediction, which is insufficient to handle bandwidth fluctuations and ensure stable playback. BONES [66] and PreSR [67] also developed SR-enhanced frameworks to optimize regular 2D video streaming. QAVA [47] presented an edge-assisted adaptive scheme for multi-user live 2D video streaming. Yet, given different viewing patterns, they struggle to directly support 360° video streaming. MA360 [68] presented a Multi-Agent DRL-based system to tackle multi-user live streaming. OmniLive [5] exploited video SR to improve 360° live streaming quality. However, they are difficult to adapt to video-on-demand streaming, focused on this work, due to the lack of long buffer consideration. Some studies [12], [58] proposed alleviating buffer restrictions using SVC, however, they do not consider the low compression ratio and cross-layer overheads associated with SVC [34].

In contrast, NETA combines the benefits of saliency analysis and long buffer to overcome the limitation of the small buffer. In addition, our previous work [30] proposed a saliency-aware and SR-enhanced 360° video streaming scheme. Despite its advances, it is still limited to the adverse effects of distortion caused by conventional codecs on SR-recovered videos. Meanwhile, it will encounter the challenges of insufficient computing resources in mobile devices. For various tiling schemes, smaller tiles attain higher viewport coverage efficiency but exhibit lower compression efficiency due to more loss of spatio-temporal correlation [13]. Therefore, TBRA [28] and VASTile [18] introduced adaptive tiling into the conventional bitrate adaptation for better bandwidth utilization. However, they increase player complexity and ignore the inherent limitations of traditional coding.

Neural video compression. DNN-based codecs (e.g., [21], [22], [39], [69]) open up new opportunities for UHD video streaming. Swift [34] and DeepWiVe [70] proposed neural-compression-enabled adaptive streaming for regular 2D videos. However, these solutions cannot directly be used for 360° videos due to the high computing requirements and distinct viewing mechanisms. By contrast, we first present a neural-compression-empowered and traditional-coding-aided streaming paradigm for 360° videos by leveraging the properties of viewport awareness and edge computing to improve the user viewing experience, adapt inadequate bandwidth, and achieve on-device resource savings.

Visual saliency. It captures the attribute of how content attracts user attention, playing a pivotal role in enhancing viewing experience [55]. Recently, visual saliency has begun to be used for 360° video services, such as viewport prediction [19], [33], [53], video compression [32] and quality adaptation [2], [30]. There are primarily two ways to acquire saliency:

the view-driven method and the content-driven approach. The former uses collected user viewing data to yield saliency maps [55], while the latter utilizes deep learning models to generate saliency maps based on the impact of video content on viewer attention [2]. In this work, we opt for the view-driven method and follow the saliency map acquisition technique proposed by [55]. Building on this, we apply tile visual saliency as priori information to proactively prefetch non-viewport tiles, effectively addressing the challenges posed by the limited viewport prediction window and fluctuating bandwidth conditions.

VII. Conclusion

We presented NETA, a neural compression-empowered and traditional coding-aided 360° video streaming paradigm. With edge assistance, we realized the best of both through implicit-explicit buffer-based prefetching grounded in visual saliency and bitrate adaptation with smart model switching around viewports, addressing critical issues in the short viewport prediction window and time-varying viewing directions. We also developed LyaDRL for joint bitrate and model adaptation, to maximize user QoE and maintain system stability.

Neural codecs rapidly evolve, introducing new opportunities and challenges for emerging streaming media, such as AI-generated content and volumetric video. Ambisonics and 3D audio play a crucial role in VR experiences. Neural codecs can also be employed for audio compression, enabling efficient transmission and audio-visual synchronization. We are also interested in cross-layer neural coding in 6G, which can directly map video signals to channel symbols, promoting a shift from data-oriented to task-oriented communication.

References

[1] H. Xiao, C. Xu, Z. Feng *et al.*, "A transcoding-enabled 360° VR video caching and delivery framework for edge-enhanced next-generation wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 40, no. 5, pp. 1615–1631, 2022.

[2] S. Wang *et al.*, "SalientVR: Saliency-driven mobile 360-degree video streaming with gaze information," in *Proceedings of the International Conference on Mobile Computing And Networking*, 2022, pp. 542–555.

[3] L. Chopra *et al.*, "Parima: Viewport adaptive 360-degree video streaming," in *Proceedings of the ACM Web Conference*, 2021, pp. 2379–2391.

[4] X. Hou, S. Dey, J. Zhang, and M. Budagavi, "Predictive adaptive streaming to enable mobile 360-degree and VR experiences," *IEEE Transactions on Multimedia*, vol. 23, pp. 716–731, 2021.

[5] S. Park, Y. Cho, H. Jun, J. Lee, and H. Cha, "OmniLive: Super-resolution enhanced 360° video live streaming for mobile devices," in *Proceedings of the Annual International Conference on Mobile Systems, Applications and Services*, 2023, pp. 261–274.

[6] VIVE Cosmos Elite, "A high-performance, all-in-one Head-Mounted Displays," https://www.vive.com/us/, 2023, [Online; accessed 20-Jan-2023].

[7] Google, "Cardboard: Experience VR in a simple way," https://arvr.google.com/cardboard/, 2022, [Online; accessed 20-Dec-2022].

[8] Q. Cheng, H. Shan, W. Zhuang *et al.*, "Design and Analysis of MEC- and Proactive Caching-Based 360° Mobile VR Video Streaming," *IEEE Transactions on Multimedia*, vol. 24, pp. 1529–1544, 2022.

[9] Y. Jin, J. Liu, F. Wang, and S. Cui, "Ebublio: Edge-assisted multiuser 360° video streaming," *IEEE Internet of Things Journal*, vol. 10, no. 17, pp. 15 408–15 419, 2023.

[10] Y. Guan *et al.*, "Pano: Optimizing 360 video streaming with a better understanding of quality perception," in *Proceedings of the ACM Special Interest Group on Data Communication*, 2019, pp. 394–407.

[11] F. Qian *et al.*, "Flare: Practical viewport-adaptive 360-degree video streaming for mobile devices," in *Proceedings of Annual International Conference on Mobile Computing and Networking*, 2018, pp. 99–114.

[12] H. Zhang, Y. Ban, Z. Guo, K. Chen, and X. Zhang, "RAM360: Robust adaptive multi-layer 360° video streaming with Lyapunov optimization," *IEEE Transactions on Multimedia*, vol. 25, pp. 4225–4239, 2023.

[13] X. Chen, T. Tan, and G. Cao, "Popularity-aware 360-degree video streaming," in *Proceedings of the IEEE Conference on Computer Communications*, 2021, pp. 1–10.

[14] X. Yuan *et al.*, "Understanding 5G performance for real-world services: a content provider's perspective," in *Proceedings of the ACM Special Interest Group on Data Communication*, 2022, pp. 101–113.

[15] T. Wiegand, G. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 560–576, 2003.

[16] G. Sullivan, J. Ohm, W. Han *et al.*, "Overview of the high efficiency video coding (HEVC) standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1649–1668, 2012.

[17] B. Bross *et al.*, "Overview of the versatile video coding (VVC) standard and its applications," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 10, pp. 3736–3764, 2021.

[18] C. Madarasingha and K. Thilakarathna, "VASTile: Viewport adaptive scalable 360-degree video frame tiling," in *Proceedings of the ACM International Conference on Multimedia*, 2021, p. 4555–4563.

[19] X. Feng *et al.*, "LiveROI: Region of interest analysis for viewport prediction in live mobile virtual reality streaming," in *Proceedings of the ACM Multimedia Systems Conference*, 2021, pp. 132–145.

[20] T. Zhao, W. Feng, H. Zeng *et al.*, "Learning-based video coding with joint deep compression and enhancement," in *Proceedings of the ACM International Conference on Multimedia*, 2022, pp. 3045–3054.

[21] J. Li, B. Li, and Y. Lu, "Hybrid spatial-temporal entropy modelling for neural video compression," in *Proceedings of the ACM International Conference on Multimedia*, 2022, pp. 1503–1511.

[22] G. Wang, J. Li, B. Li, and Y. Lu, "EVC: Towards real-time neural image compression with mask decay," in *Proceedings of the International Conference on Learning Representations*, 2023.

[23] T. Tan and G. Cao, "Deep learning on mobile devices through neural processing units and edge computing," in *Proceedings of the IEEE Conference on Computer Communications*, 2022, pp. 1209–1218.

[24] S. Xie, Y. Xue, Y. Zhu, and Z. Wang, "Cost effective MLaaS federation: A combinatorial reinforcement learning approach," in *Proceedings of the IEEE Conference on Computer Communications*, 2022, pp. 2078–2087.

[25] Valve Corporation, "SteamVR: An ultimate tool for experiencing VR content on the Head-Mounted Displays," https://www.steamvr.com/, 2024, [Online; accessed 18-Feb-2024].

[26] Meta, "Oculus: An all-in-One VR platform and storefront." https://www.oculus.com/, 2024, [Online; accessed 18-Feb-2024].

[27] Y. Zhang, P. Zhao, K. Bian *et al.*, "DRL360: 360-degree video streaming with deep reinforcement learning," in *Proceedings of the IEEE Conference on Computer Communications*, 2019, pp. 1252–1260.

[28] L. Zhang, Y. Suo, X. Wu *et al.*, "TBRA: Tiling and bitrate adaptation for mobile 360-degree video streaming," in *Proceedings of the ACM International Conference on Multimedia*, 2021, pp. 4007–4015.

[29] K. Spiteri, R. Urgaonkar, and R. K. Sitaraman, "BOLA: Near-optimal bitrate adaptation for online videos," *IEEE/ACM Transactions On Networking*, vol. 28, no. 4, pp. 1698–1711, 2020.

[30] J. Shi *et al.*, "Sophon: Super-resolution enhanced 360° video streaming with visual saliency-aware prefetch," in *Proceedings of the ACM International Conference on Multimedia*, 2022, pp. 3124–3133.

[31] H. Yeo, Y. Jung, J. Kim *et al.*, "Neural adaptive content-aware internet video delivery," in *Proceedings of the USENIX Symposium on Operating Systems Design and Implementation*, 2018, pp. 645–661.

[32] D. Baek *et al.*, "Sali360: Design and implementation of saliency based video compression for 360° video streaming," in *Proceedings of the ACM Multimedia Systems Conference*, 2020, pp. 141–152.

[33] M. Qiao and others., "Viewport-dependent saliency prediction in 360° video," *IEEE Transactions on Multimedia*, vol. 23, pp. 748–760, 2021.

[34] M. Dasari, K. Kahatapitiya *et al.*, "Swift: Adaptive video streaming with layered neural codecs," in *Proceedings of the USENIX Symposium on Networked Systems Design and Implementation*, 2022, pp. 103–118.

[35] L. Xie *et al.*, "360ProbDASH: Improving QoE of 360 video streaming using tile-based http adaptive streaming," in *Proceedings of the ACM International Conference on Multimedia*, 2017, p. 315–323.

[36] M. Dasari, A. Bhattacharya, S. Vargas *et al.*, "Streaming 360-degree videos using super-resolution," in *Proceedings of the IEEE Conference on Computer Communications*, 2020, pp. 1977–1986.

[37] W. Zhang, F. Qian, B. Han, and P. Hui, "Deepvista: 16k panoramic cinema on your mobile device," in *Proceedings of the ACM Web Conference*, 2021, pp. 2232–2244.

[38] GPAC, "Streaming HEVC tiled 360° DASH videos," https://github.com/gpac/gpac/wiki/Tiled-Streaming, 2022, [Online; accessed 15-Jan-2023].
[39] G. Lu, W. Ouyang, D. Xu *et al.*, "DVC: An end-to-end deep video compression framework," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 11 006–11 015.
[40] A. Mercat, M. Viitanen, and J. Vanne, "UVG dataset: 50/120fps 4K sequences for video codec analysis and development," in *Proceedings of the ACM Multimedia Systems Conference*, 2020, pp. 297–302.
[41] L. Wang, M. Hajiesmaili, and R. K. Sitaraman, "FOCAS: Practical video super resolution using foveated rendering," in *Proceedings of the ACM International Conference on Multimedia*, 2021, pp. 5454–5462.
[42] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.
[43] Z. Wang, E. P. Simoncelli, and A. C. Bovik, "Multiscale structural similarity for image quality assessment," in *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*, vol. 2, 2003, pp. 1398–1402.
[44] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 586–595.
[45] X. Li, J. Liu, S. Wang *et al.*, "Efficient meta-tuning for content-aware neural video delivery," in *Proceedings of the European Conference on Computer Vision*, 2022, p. 308–324.
[46] Z. Zhu, X. Feng, Z. Tang, N. Jiang *et al.*, "Power-efficient live virtual reality streaming using edge offloading," in *Proceedings of the ACM Workshop on Network and Operating Systems Support for Digital Audio and Video*, 2022, pp. 57–63.
[47] X. Ma *et al.*, "QAVA: QoE-aware adaptive video bitrate aggregation for HTTP live streaming based on smart edge computing," *IEEE Transactions on Broadcasting*, vol. 68, no. 3, pp. 661–676, 2022.
[48] Y. Lu, Y. Zhu, and Z. Wang, "Personalized 360-degree video streaming: A meta-learning approach," in *Proceedings of the ACM International Conference on Multimedia*, 2022, p. 3143–3151.
[49] X. Yang, H. Lin, Z. Li *et al.*, "Mobile access bandwidth in practice: Measurement, analysis, and implications," in *Proceedings of the ACM Special Interest Group on Data Communication*, 2022, pp. 114–128.
[50] N. Li and Y. Liu, "EVASR: Edge-based video delivery with salience-aware super-resolution," in *Proceedings of the ACM Multimedia Systems Conference*, 2023, pp. 142–152.
[51] B. Abolhassani, J. Tadrous, and A. Eryilmaz, "Single vs distributed edge caching for dynamic content," *IEEE/ACM Transactions on Networking*, vol. 30, no. 2, pp. 669–682, 2021.
[52] D. Xu, T. Li, Y. Li, X. Su, S. Tarkoma, T. Jiang, J. Crowcroft, and P. Hui, "Edge intelligence: Empowering intelligence to the edge of network," *Proceedings of the IEEE*, vol. 109, no. 11, pp. 1778–1837, 2021.
[53] J. Chen, X. Luo, M. Hu, D. Wu, and Y. Zhou, "Sparkle: User-aware viewport prediction in 360-degree video streaming," *IEEE Transactions on Multimedia*, vol. 23, pp. 3853–3866, 2021.
[54] Federal communications commission (FCC), "Measuring broadband raw data releases," https://www.fcc.gov/oet/mba/raw-data-releases/, 2023, [Online; accessed 3-Jan-2023].
[55] A. Nguyen *et al.*, "A saliency dataset for 360-degree videos," in *Proceedings of the Multimedia Systems Conference*, 2019, pp. 279–284.
[56] C. Wu, Z. Tan, Z. Wang, and S. Yang, "A dataset for exploring user behaviors in VR spherical video streaming," in *Proceedings of the ACM Multimedia Systems Conference*, 2017, pp. 193–198.
[57] M. Palash, V. Popescu, A. Sheoran, and S. Fahmy, "Robust 360° video streaming via non-linear sampling," in *Proceedings of the IEEE Conference on Computer Communications*, 2021, pp. 1–10.
[58] Y. Xie, Y. Zhang, and T. Lin, "Deep curriculum reinforcement learning for adaptive 360° video streaming with two-stage training," *IEEE Transactions on Broadcasting*, 2023.
[59] S. Bi, L. Huang, H. Wang, and Y. A. Zhang, "Lyapunov-guided deep reinforcement learning for stable online computation offloading in mobile-edge computing networks," *IEEE Transactions on Wireless Communications*, vol. 20, no. 11, pp. 7519–7537, 2021.
[60] J. Yu, Y. Li, X. Liu *et al.*, "IRS assisted NOMA aided mobile edge computing with queue stability: Heterogeneous multi-agent reinforcement learning," *IEEE Transactions on Wireless Communications*, 2022.
[61] D. Raca, D. Leahy *et al.*, "Beyond throughput, the next generation: A 5G dataset with channel and context metrics," in *Proceedings of the ACM Multimedia Systems Conference*, 2020, p. 303–308.
[62] FFmpeg, "A complete and cross-platform solution to record and convert video," https://ffmpeg.org, 2022, [Online; accessed 18-Dec-2022].
[63] D. Wu, P. Wu, M. Zhang, and F. Wang, "MANSY: Generalizing neural adaptive immersive video streaming with ensemble and representation learning," *arXiv preprint arXiv:2311.06812*, 2023.
[64] C. Zheng, J. Yin, F. Wei, Y. Guan, Z. Guo, and X. Zhang, "STC: FoV tracking enabled high-quality 16K VR video streaming on mobile platforms," *IEEE Transactions on Circuits and Systems for Video Technology (TCSVT)*, vol. 32, no. 4, pp. 2396–2410, 2021.
[65] H. Hu, Z. Xu, X. Zhang, and Z. Guo, "Optimal viewport-adaptive 360-degree video streaming against random head movement," in *IEEE International Conference on Communications*, 2019, pp. 1–6.
[66] L. Wang *et al.*, "BONES: Near-optimal neural-enhanced video streaming," *arXiv preprint arXiv:2310.09920*, 2023.
[67] G. Zhou, Z. Luo, M. Hu, and D. Wu, "PreSR: Neural-enhanced adaptive streaming of VBR-encoded videos with selective prefetching," *IEEE Transactions on Broadcasting*, vol. 69, no. 1, pp. 49–61, 2022.
[68] Y. Ban, Y. Zhang, H. Zhang, X. Zhang, and Z. Guo, "MA360: Multi-agent deep reinforcement learning based live 360-degree video streaming on edge," in *IEEE International Conference on Multimedia and Expo*, 2020, pp. 1–6.
[69] X. Sheng, J. Li, B. Li *et al.*, "Temporal context mining for learned video compression," *IEEE Transactions on Multimedia*, 2022.
[70] T. Tung and D. Gündüz, "DeepWiVe: Deep-learning-aided wireless video transmission," *IEEE Journal on Selected Areas in Communications*, vol. 40, no. 9, pp. 2570–2583, 2022.

Jianxin Shi is currently a Ph.D. student at Nankai University, Tianjin, China. He is also currently a joint Ph.D. student at Simon Fraser University, British Columbia, Canada. His research interests include neural-enhanced video streaming, volumetric content processing and communications, and satellite networking.

Miao Zhang received her B.Eng. degree from Sichuan University in 2015, and her M.Eng. degree from Tsinghua University in 2018. She is currently a Ph.D. student at Simon Fraser University, British Columbia, Canada. Her research areas include cloud and edge computing, and multimedia systems and applications.

Linfeng Shen is currently a Ph.D. student in the School of Computing Science at Simon Fraser University, BC, Canada. He received BEng in information security from Beijing University of Posts and Telecommunications in 2019, and MSc in computing science from Simon Fraser University in 2021. His research interests include edge computing and multimedia.

Jiangchuan Liu (S'01-M'03-SM'08-F'17) is a University Professor in the School of Computing Science, Simon Fraser University, British Columbia, Canada. He is a Fellow of The Canadian Academy of Engineering, an IEEE Fellow, and an NSERC E.W.R. Steacie Memorial Fellow. He was an EMC-Endowed Visiting Chair Professor of Tsinghua University (2013-2016). In the past, he worked as an Assistant Professor at The Chinese University of Hong Kong and as a research fellow at Microsoft Research Asia. He received the BEng degree (cum laude) from Tsinghua University, Beijing, China, in 1999, and the PhD degree from The Hong Kong University of Science and Technology in 2003, both in computer science. He is a co-recipient of the inaugural Test of Time Paper Award of IEEE INFOCOM (2015), ACM SIGMM TOMCCAP Nicolas D. Georganas Best Paper Award (2013), and ACM Multimedia Best Paper Award (2012). His research interests include multimedia systems and networks, cloud and edge computing, social networking, online gaming, and Internet of things/RFID/backscatter. He has served on the editorial boards of IEEE/ACM Transactions on Networking, IEEE Transactions on Big Data, IEEE Transactions on Multimedia, IEEE Communications Surveys and Tutorials, and IEEE Internet of Things Journal. He is a Steering Committee member of IEEE Transactions on Mobile Computing and Steering Committee Chair of IEEE/ACM IWQoS (2015-2017). He is TPC Co-Chair of IEEE INFOCOM'2021 and General Co-Chair of INFOCOM'2024.

Lingjun Pu received the Ph.D. degree from Nankai University in 2016. He is currently an associate professor at Nankai University, Tianjin, China. He was a joint PhD student with the University of Göttingen, Germany, from 2013 to 2015. His current research interests include programmable networks, edge intelligence, UHD video streaming, and resource scheduling.

Jingdong Xu is a full professor at Nankai University, Tianjin, China. She is the head with the Computer Networks and Information Security Lab. Her research interests include mobile computing, network security, Internet of Things, and blockchain.