EOS: Energy-Optimized Super-Resolution on Mobile Devices for Live 360-Degree Videos

Seonghoon Park
Yonsei University
Seoul, Republic of Korea
park.s@yonsei.ac.kr

Minchan Kim
Yonsei University
Seoul, Republic of Korea
mc.kim@yonsei.ac.kr

Hyejin Park
Yonsei University
Seoul, Republic of Korea
hyejin.park@yonsei.ac.kr

Jeho Lee
Yonsei University
Seoul, Republic of Korea
jeholee@yonsei.ac.kr

Jiwon Kim
Uppsala University
Uppsala, Sweden
jiwon.kim@angstrom.uu.se

Hojung Cha*
Yonsei University
Seoul, Republic of Korea
hjcha@yonsei.ac.kr

Abstract

Although on-device video super-resolution enables high-quality live 360-degree streaming on mobile devices, existing methods often waste energy by overlooking perceived visual quality. In this paper, we present EOS, an energy-efficient on-device super-resolution system for mobile omnidirectional video (ODV) live streaming. EOS reduces energy waste by dynamically adjusting super-resolution complexity based on the predicted visual quality of super-resolved frames. This approach raises two challenges: (1) designing an adaptive inference policy that maximizes energy savings while minimizing degradation in Quality-of-Experience (QoE), and (2) developing a method to predict visual quality under the constraints of mobile ODV live streaming. To tackle these challenges, EOS introduces EOS SR and a No-Reference Upscaling Quality Prediction scheme. EOS SR employs a device-agnostic, scalable deep neural network optimized for mobile devices, with an energy-aware scheduler that jointly selects the optimal super-resolution model and GPU frequency. The No-Reference Upscaling Quality Prediction scheme estimates visual quality across arbitrary viewpoints in real time without requiring high-resolution reference videos. Experiments on commodity smartphones show that EOS reduces average power consumption by 34.6%–49.9% compared to baseline methods, while preserving high visual quality and frame rates.

*Corresponding author

ACM MOBICOM '25, Hong Kong, China

ACM ISBN 979-8-4007-1129-9/2025/11
https://doi.org/10.1145/3680207.3765252

CCS Concepts

• **Human-centered computing** → **Mobile computing**; **Mobile devices**; • **Computing methodologies** → **Computer vision**.

Keywords

360-degree videos, omnidirectional videos, live streaming, video super-resolution, power management

ACM Reference Format:
Seonghoon Park, Minchan Kim, Hyejin Park, Jeho Lee, Jiwon Kim, and Hojung Cha. 2025. EOS: Energy-Optimized Super-Resolution on Mobile Devices for Live 360-Degree Videos. In *The 31st Annual International Conference on Mobile Computing and Networking (ACM MOBICOM '25), November 4–8, 2025, Hong Kong, China.* ACM, New York, NY, USA, 16 pages. https://doi.org/10.1145/3680207.3765252

1 Introduction

Mobile omnidirectional video (ODV) live streaming, which provides immersive and interactive user experiences through live 360-degree videos, has gained significant attention as it can be extended to emerging mobile applications such as extended reality (XR). Unfortunately, unlike standard 2D videos that offer a single fixed viewpoint, ODVs, which encompass views in all 360-degree directions, face challenges in delivering high-resolution live videos over mobile networks. For instance, achieving a viewport resolution of 1280×720 in an ODV requires an overall resolution of 5120×2560, which demands tens of Mbps in network bandwidth, making it difficult to transmit over current mobile networks.

On-device video super-resolution is a recent approach to addressing the challenge of delivering high-resolution ODVs to users in mobile ODV streaming. This technique uses deep neural networks (DNNs) to upscale low-resolution videos transmitted to mobile devices into high-resolution videos in real time. Notably, a recent study, OmniLive [43], introduced an approach that maximizes the use of a smartphone's

GPU resources to enhance the upscaling quality of the super-resolution DNN. This solution employs adaptive inference to ensure that the super-resolution process consistently completes while making full use of the available time budget, regardless of the device type or state. For a 30 frames-per-second (FPS) video, the time budget corresponds to 33 ms per frame. However, the approach of maximizing GPU usage for DNNs can lead to excessive and unnecessary GPU consumption, thereby rapidly draining the smartphone's battery. Our preliminary experiments demonstrate that even a lightweight super-resolution DNN can meet specific upscaling quality standards for certain video frames. These experimental results suggest that existing methods, by not considering the visual quality perceived by the end user, often result in excessive DNN inference and thus lead to energy inefficiency.

In this paper, we aim to develop an energy-efficient on-device video super-resolution system for mobile ODV live streaming. There are two key technical challenges in designing such a system. The first challenge is the need for an energy-efficient inference policy that accommodates the computational diversity of smartphones while ensuring no loss in quality of experience (QoE). The primary QoE factors during video playback are frame rates and user-perceived visual quality. Given the varying computational capacities across devices, it is not trivial to develop an adaptive inference technique that effectively balances frame rates, user-perceived visual quality, and energy consumption. The second challenge is to predict the user-perceived visual quality of upscaled videos at runtime using only low-resolution videos. VMAF [5], which is known for accurately reflecting users' perceptual experiences, requires high-resolution reference data for its calculation. This presents a challenge, as high-resolution ODV cannot be transmitted due to uplink bandwidth constraints in mobile ODV live streaming. Moreover, ODVs have an infinite number of possible viewpoints, making it extremely difficult to predict VMAF scores for all these scenarios in real time.

Addressing these challenges, we propose EOS, an energy-optimized super-resolution system for mobile ODV live streaming that does not compromise perceived visual quality or frame rates. To tackle the challenge of designing an energy-efficient inference policy, EOS introduces EOS SR. EOS SR consists of EOSNet and the EOS SR Scheduler. EOSNet is a device-agnostic and scalable DNN model for on-device video super-resolution. The EOS SR Scheduler combines DNN model selection with GPU frequency scaling to minimize power consumption by jointly considering the expected upscaling quality and the device's computational capacity. To address the challenge of runtime visual quality prediction, EOS features No-Reference Upscaling Quality Prediction, which predicts VMAF scores for super-resolved videos across arbitrary viewpoints in real time without requiring high-resolution reference videos. The streaming server utilizes a deep learning-based VMAF map predictor to extract features from low-resolution ODVs in equirectangular projection (ERP) format and generate VMAF maps for various viewpoints. The client device then references these maps to obtain fine-grained VMAF scores for specific viewpoints.

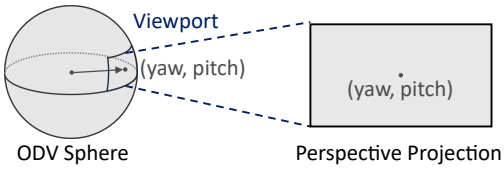


Figure 1: ODV projection.

We implemented the complete EOS system, spanning from the streaming server to the Android application. In real mobile ODV live streaming scenarios, our experiments demonstrated that EOS reduced average power consumption by 34.59% to 49.94% on commodity smartphones, compared to baseline methods, without sacrificing QoE.

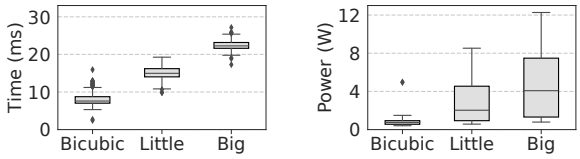
2 Background

This section discusses mobile ODV live streaming and the need for video super-resolution.

2.1 Mobile ODV Live Streaming

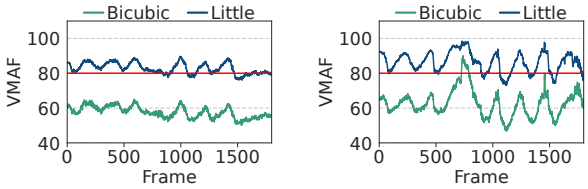
Mobile ODV live streaming delivers live videos recorded with 360-degree cameras to viewers over mobile networks. Unlike 2D videos that provide only a fixed viewpoint, ODVs—also known as 360-degree or panoramic videos—offer viewpoints in all directions centered around the camera, enabling a more immersive and interactive experience. For instance, through mobile ODV live streaming, streamers can share real-time travel moments or broadcast sports events, allowing viewers to explore multiple viewpoints. This mobile ODV live streaming can also be extended to XR applications.

Figure 1 illustrates the process of perspective projection, which transforms the entire spherical view of a 360-degree image into a two-dimensional image based on a specific viewpoint. In this process, the viewpoint is defined by yaw, which ranges from -180° to 180°, and pitch, which ranges from -90° to 90°. Yaw and pitch represent the horizontal and vertical rotation angles in the spherical coordinate system, respectively. The horizontal and vertical fields of view determine the range of the scene captured in the projected image relative to the viewpoint. Using these parameters, perspective projection is applied to naturally convert the spherical view into a flat image. The number of possible perspective projection images that can be generated from a single omnidirectional image is virtually limitless. If the yaw and pitch are adjusted by 1 degree each, there are 361 possible values for yaw and 181 possible values for pitch, resulting in 65,341



(a) Inference time (b) Power consumption

Figure 2: Comparison between upscaling methods.



(a) View 1 (b) View 2

Figure 3: VMAF traces.

possible viewpoints. As video players allow fine control of viewpoints, the total number of viewpoints can range from hundreds of thousands to millions.

2.2 Need for Video Super-Resolution

Mobile ODV live streaming faces the challenge of delivering high-quality viewing experiences to users. In 2D video streaming, the video being transmitted directly corresponds to the portion displayed in the viewer's device viewport. On the other hand, ODV streaming requires the transmission of not only the portion displayed in the viewport but also parts corresponding to other viewing angles. For example, while 2D video streaming with a viewport resolution of 1280×720 requires about 4-5 Mbps, ODV streaming demands approximately 100 Mbps [43]. Unfortunately, in mobile network environments, uplink bandwidth often does not exceed 10 Mbps [7], limiting mobile ODV streaming to delivering lower-resolution videos to users.

A recent approach to addressing the challenge of delivering high-quality video to viewers in mobile ODV live streaming is the use of video super-resolution. Video super-resolution is a technique that utilizes deep learning to upscale low-resolution videos into high-resolution ones. This deep learning-based approach provides better upscaling quality compared to traditional methods such as nearest, bilinear, or bicubic interpolation, and has therefore been actively employed in video streaming [11, 20, 58, 59]. As the GPU capabilities of mobile devices have improved, there have been many efforts to perform video super-resolution tasks on smartphones [29, 40, 57]. In particular, OmniLive [43] introduced video super-resolution to address the low-resolution issue in mobile ODV live streaming. This study tried to maximize the upscaling quality of super-resolution models by leveraging as much of the smartphone's GPU resources as possible, without degrading frame rates such as 30 FPS.

3 Understanding Energy Inefficiency

Since smartphones have limited battery capacity, optimizing energy consumption to maximize battery lifetime has long been a critical research issue [16, 18, 19, 30, 33, 36, 37, 39]. However, the recent video super-resolution solution for mobile ODV live streaming can lead to excessive energy consumption because the solution utilizes the maximum possible GPU resources. If the target upscaling quality for super-resolution is predefined and GPU resources are used only to the extent necessary to meet the target quality, unnecessary energy usage could be reduced. For example, VMAF [5], which is known for effectively reflecting subjective user experiences, considers a score of 80 or higher as good quality [10, 38]. This suggests that the target visual quality for super-resolution can be set to a VMAF score of 80.

To examine the energy inefficiency of the existing solution, we conducted experiments measuring the power consumption and VMAF scores of upscaled frames across various upscaling methods. In this experiment, we compared three upscaling methods: bicubic interpolation, a little DNN model, and a big DNN model. The task involved 4× upscaling, which enhances video frames from a resolution of 320×180 to 1280×720. The big model is a super-resolution model designed to maximize GPU resource usage as closely as possible to the 30 FPS constraint on the target smartphone. The little model is a simplified version of the big DNN model, with smaller block channels and fewer neural blocks. The smartphone used in this experiment was a Pixel 6 Pro, equipped with an ARM Mali-G78 MP20 858 MHz GPU. We used an ODV recorded at a resolution of 5120×2560 and 30 FPS for 1 minute with an Insta360 X3 360-degree camera [2]. The ODV was converted into perspective-projected videos with two fixed viewpoints, View 1 and View 2. The (yaw, pitch) values for View 1 and View 2 were (0°, 0°) and (0°, 45°), respectively. The resolution of the perspective-projected video was 1280×720. Based on these three upscaling methods and the two viewpoint videos, we measured inference time, power consumption, and VMAF scores. During the measurement, the GPU frequency of the smartphone was fixed at the maximum value of 858 MHz.

Figure 2 presents the average inference time and power consumption measured across the upscaling methods. Notably, the little model, with an average inference time of 15.1 ms, demonstrated a 32.8% reduction in inference time compared to the big model, which averaged 22.5 ms. As a result, the little model exhibited an average power consumption of 2.9 W, reducing the average power consumption by

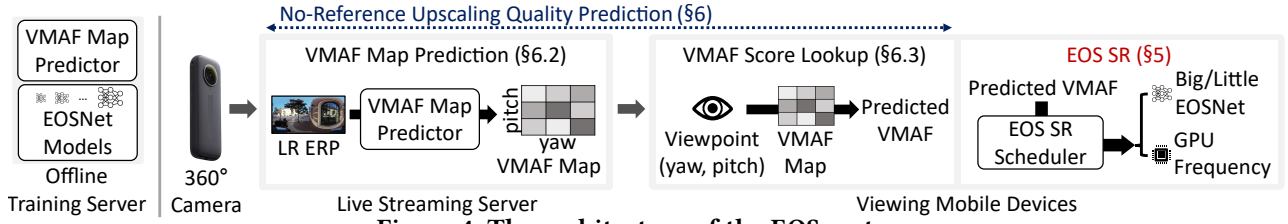


Figure 4: The architecture of the EOS system.

38.4% compared to the big model, which consumed 4.7 W on average. This reduction in power consumption is attributed to the decreased GPU utilization corresponding to the shorter inference time. Figure 3 shows the VMAF traces of the little model and bicubic interpolation. As indicated in the figures, although bicubic interpolation exhibited significantly lower inference time and power usage compared to the other methods, it resulted in notably low VMAF scores, suggesting that it is not a viable method for high-quality upscaling. In contrast, the little model often achieved VMAF scores exceeding 80, indicating good visual quality. These results suggest that the previous approach of relying solely on the big model to maximize inference time can lead to an overutilization of computational resources, thereby resulting in energy inefficiency.

4 EOS Overview

We discuss the technical challenges involved in energy optimization and introduce EOS, an energy-efficient super-resolution system designed to address the challenges.

4.1 Challenges

To optimize the energy consumption of the on-device super-resolution system for mobile ODV live streaming, two technical challenges must be addressed. The first is how to design an inference policy that minimizes QoE loss while optimizing energy consumption. The key QoE metrics related to video playback are visual quality and frame rates. While more complex models ensure higher visual quality, they come at the expense of energy efficiency. In contrast, less complex models improve energy efficiency but result in lower visual quality. Therefore, an inference policy must be able to select a DNN model with appropriate computational complexity to balance visual quality and energy efficiency. Meanwhile, to prevent frame rate loss, the inference time of the super-resolution model should not exceed certain time constraints. For example, 30-FPS videos allow only 33 ms of computation time per frame, but the inference time for the same super-resolution model can vary depending on the mobile devices, whose GPU capabilities are diverse. Therefore, balancing frame rates, visual quality, and energy efficiency remains a significant challenge.

The second challenge is predicting the visual quality of super-resolved videos in advance to avoid unnecessary computational resource usage in super-resolution tasks. VMAF, which is known for accurately reflecting subjective user satisfaction, is a full-reference metric that requires an undistorted reference video for its measurements. Unfortunately, as discussed in Section 2.2, in mobile ODV live streaming, the limitations of mobile network uplink bandwidth prevent the transmission of high-resolution reference videos to the streaming server. Furthermore, due to the spherical nature of ODV, there are a vast number of possible viewpoints users can select while watching videos, as discussed in Section 2.1. Predicting the upscaling quality for such a wide range of viewpoints without a reference video, and within a limited time frame, presents a significant technical challenge.

4.2 Overall Architecture

We propose EOS, an energy-optimized on-device super-resolution system for mobile ODV live streaming. To address the challenges discussed in Section 4.1, EOS introduces EOS SR and No-Reference Upscaling Quality Prediction, which we discuss in detail in Sections 5 and 6, respectively. As discussed in Section 3, target visual quality is set to a VMAF score of 80. Also, the target frame rate is set to 30 FPS, which is a widely used standard in mobile video applications [31, 34, 43, 57]. The policy of EOS is generalizable and can operate consistently even when alternative quality metrics or target values are adopted.

Figure 4 shows the overall architecture of the EOS system, including an offline training server, a live streaming server, and viewing mobile devices. The offline training server trains EOSNet and the VMAF map predictor, which are detailed in Sections 5.1 and 6.2, respectively. Under uplink bandwidth constraints, the streaming server receives low-resolution live ODV streams captured by 360-degree cameras and transmits them to the viewing mobile devices of multiple users. On the streaming server, the No-Reference Upscaling Quality Prediction takes a single low-resolution ODV frame as input to simultaneously predict the visual quality of super-resolved video across a wide range of viewing angles. The viewing mobile device estimates the VMAF score for the given viewing angle using the VMAF map transmitted from the streaming server. The device plays the live ODV streams received

from the streaming server by applying perspective projection based on the viewing angle. Based on the predicted VMAF score, the system performs energy-efficient super-resolution through EOS SR, and then the upscaled video frames are displayed on the device.

5 EOS SR

EOS SR is an inference technique for energy-efficient on-device video super-resolution without QoE degradation. We propose a super-resolution DNN model for EOS SR called EOSNet and introduce the EOS SR scheduler, an on-device video super-resolution policy based on EOSNet.

5.1 EOSNet

EOSNet is a device-agnostic and scalable DNN model for video super-resolution, specifically designed for EOS SR. The model is based on the OmniSRNet architecture [43], but in terms of generality, EOSNet and OmniSRNet differ significantly. While OmniSRNet employs neural architecture search to find the optimal neural network for each type of mobile device, EOSNet supports general adaptive inference across various devices with a single device-agnostic model. Optimizing a separate neural network for each device would result in an impractically large number of cases to predict for the No-Reference Upscaling Quality Prediction, which is discussed in Section 6.

Figure 5 illustrates the structure of EOSNet. This model takes a low-resolution image as input, extracts image features through multiple dense blocks [28], and reconstructs the image using subpixel upscaling [12]. EOS SR manages video super-resolution by upscaling each frame through EOSNet. As shown in the figure, EOSNet enables adaptive inference by configuring the parameters of the dense blocks responsible for feature extraction. We represent the EOSNet model as $M^{n,c}$, where n refers to the number of dense blocks and c denotes the channel size of each block. Increasing n enhances the accuracy of feature extraction but also increases computational overhead. EOSNet selects n from the set of natural numbers between 1 and 4. Since relying solely on the number of dense blocks may not provide sufficient precision for adaptive inference, EOSNet further adjusts the channel size c to achieve more refined adaptability. However, EOSNet maintains a constant c across all dense blocks during a single inference. The value of c is defined as 2^k, where k is a natural number not exceeding 6. Additionally, to maintain smooth gradient flow and reduce information loss, the channel size of the bottleneck block within each dense block—responsible for concatenating the outputs of previous layers—is set to $12 \times (1 + n)$. This model architecture is both general and scalable, allowing it to be expanded to larger values of n and c as smartphone GPU performance improves.

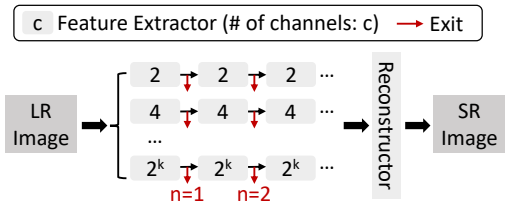


Figure 5: The EOSNet architecture.

5.2 EOS SR Scheduler

The EOS SR scheduler is an on-device video super-resolution policy that operates based on EOSNet. The EOS SR scheduler incorporates a big/little model selection scheme and GPU frequency scaling to select the optimal EOSNet model and GPU frequency for each frame. This policy minimizes the loss of upscaling quality and frame rates while performing energy-efficient super-resolution.

Big/little model selection. Big/little model selection refers to a scheme that selects the appropriate model between an energy-efficient little model and resource-intensive big models. The little model is fixed as $M^{1,2}$, the lightest model of EOSNet, across all types of mobile devices. This model performs super-resolution with minimal computation. The big models include all other $M^{n,c}$ models except for $M^{1,2}$. Since the inference time of $M^{n,c}$ varies depending on the GPU performance of the device, and the inference deadline changes based on the device's computational load, the EOS SR scheduler dynamically selects the largest $M^{n,c}$ that satisfies the constraints at runtime. The time limit $T_{\text{threshold}}$ for the super-resolution of one frame by EOSNet is determined as follows:

$$T_{\text{threshold}} = T_{\text{deadline}} - T_{\text{others}}. \quad (1)$$

T_{deadline} is the minimum time required to process one frame to play the video within the given frame rates, e.g., 33 ms for 30 FPS, and T_{others} represents the time taken for tasks that must be completed prior to the super-resolution task, such as processing the 360-degree image or reading data from memory. T_{others} varies slightly depending on the CPU or memory conditions. The EOS SR scheduler measures the time consumed immediately prior to the super-resolution task for each frame and performs the super-resolution task within the remaining time budget, $T_{\text{threshold}}$.

Algorithm 1 outlines the detailed operation of the big/little model selection scheme. This algorithm takes $Q_{\text{predicted}}$, $Q_{\text{threshold}}$, $T_{\text{threshold}}$ as input. $Q_{\text{predicted}}$ is the predicted user-perceived video quality of the current frame when using the little model, $Q_{\text{threshold}}$ is the quality threshold for determining the use of the little model, and $T_{\text{threshold}}$ is the inference time threshold required to meet the frame rate constraints. Note that the target quality metric of big/little model selection is VMAF. As mentioned in Section 3, a VMAF score of 80 is generally considered good quality, and hence $Q_{\text{threshold}}$ is set to

Algorithm 1 Big/Little Model Selection.

Input: $Q_{\text{predicted}}$, $Q_{\text{threshold}}$, $T_{\text{threshold}}$
Output: Selected EOSNet model M_{out}

1: **if** $Q_{\text{predicted}} > Q_{\text{threshold}}$ **then**
2: $n_{\text{out}}, c_{\text{out}} \leftarrow 1, 2$ ▷ little model
3: **else**
4: **for all** (n, c) in descending order **do**
5: $T_{\text{sr}} \leftarrow T_{\text{input}} + n_i \cdot T^{c_i}_{\text{feature}} + T_{\text{reconstructor}}$
6: **if** $T_{\text{sr}} < T_{\text{threshold}}$ **then**
7: $n_{\text{out}}, c_{\text{out}} \leftarrow n_i, c_i$ ▷ big model
8: **break**
9: $M_{\text{out}} \leftarrow M^{n_{\text{out}}, c_{\text{out}}}$

80. If $Q_{\text{predicted}}$ exceeds $Q_{\text{threshold}}$, the little model is selected because the model can deliver sufficient user-perceived quality (Lines 1–2). If the little model is insufficient to provide the target upscaling quality, one of the big models suited to the device's resource conditions is selected.

When selecting the big model, the super-resolution processing time T_{sr} is estimated for each model, starting from the most complex EOSNet model (Line 4). Here, n_i and c_i refer to the specific n and c values for the currently selected model. Note that the EOSNet model consists of an input layer, feature extractors, and reconstructors. The computation times for the input layer and reconstructor, T_{input} and $T_{\text{reconstructor}}$, are the same across all models. For the feature extractor, computation time varies with the channel size c of the dense block, and T^{c}_{feature} represents the computation time for one block with the given c. Since the total computation time for the feature extractor is proportional to the number of dense blocks n, the time becomes $n \cdot T^{c}_{\text{feature}}$. The algorithm assumes T_{input}, T^{c}_{feature}, and $T_{\text{reconstructor}}$ are pre-measured at the maximum GPU frequency for each device. Using these values, the algorithm calculates T_{sr} (Line 5). If T_{sr} is less than $T_{\text{threshold}}$, the current n_i and c_i are selected as parameters for the big model (Lines 6–8).

GPU frequency scaling. GPU frequency scaling is a scheme that determines the most energy-efficient GPU frequency to operate the selected EOSNet model. When a lighter EOSNet model is chosen, lowering the GPU frequency can further reduce power consumption. Previous research [43] has shown that GPU utilization during ODV playback is minimal. Thus, dynamically scaling the GPU frequency for the super-resolution task is expected to have a negligible impact on other background GPU workloads.

The function $T_{\text{sr}}(f)$, which calculates the inference time of the selected EOSNet at an arbitrary GPU frequency f, is defined as follows:

$$T_{\text{sr}}(f) = T_{\text{sr}}(f_{\max}) \cdot \frac{f_{\max}}{f}, \tag{2}$$

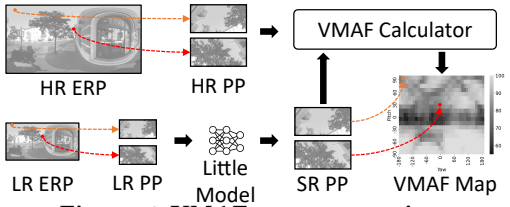


Figure 6: VMAF map generation.

where $f_{\max}$ is the maximum GPU frequency of the device. $T_{\text{sr}}(f_{\max})$ is the inference time of the selected EOSNet at $f_{\max}$. Since Algorithm 1 assumes the GPU frequency is at its maximum, $T_{\text{sr}}(f_{\max})$ can be obtained in the same way as in line 5 of the algorithm.

Based on $T_{\text{sr}}(f_{\max})$, the most energy-efficient GPU frequency for the current EOSNet, f_{optimal}, is selected. f_{optimal} is the lowest GPU frequency among those that do not cause a drop in frame rates, as shown below:

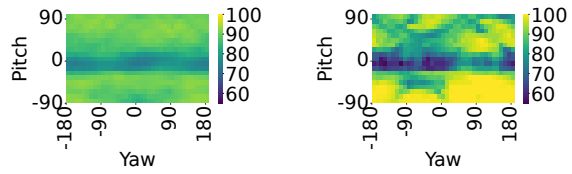
$$f_{\text{optimal}} = \min\{f \mid T_{\text{sr}}(f) \leq T_{\text{threshold}}\}. \tag{3}$$

6 No-Reference Upscaling Quality Prediction

The No-Reference Upscaling Quality Prediction estimates the visual quality of super-resolved frames, which is used by the EOS SR Scheduler. As discussed in Section 2.2, in mobile ODV live streaming, transmitting high-resolution videos, which act as a ground-truth reference, to the server is infeasible due to uplink bandwidth limitations. Therefore, the approach involves predicting VMAF maps using only the available low-resolution ODVs. This section discusses the process of generating the VMAF map and predicting VMAF scores based on the map.

6.1 VMAF Map Generation

As discussed in Section 4.1, the No-Reference Upscaling Quality Prediction must be able to predict VMAF scores for numerous viewpoints in real time. One possible approach is to generate perspective-projected images for each viewpoint and then predict the upscaling quality of those images. However, this approach is not feasible in a live-streaming environment due to the significant time overhead caused by the vast number of viewpoints. Therefore, No-Reference Upscaling Quality Prediction adopts an approach that takes a low-resolution ODV frame in ERP format as input and predicts the upscaling quality for multiple viewpoints in a single operation. The upscaling quality for different viewpoints is managed in a data structure called a VMAF map. The VMAF map is a two-dimensional array that contains VMAF scores for various viewpoints, defined by different yaw and pitch angles at regular intervals.



(a) General pattern (b) Specific pattern

Figure 7: VMAF scores depending on yaw and pitch.

Figure 6 illustrates the process of generating the VMAF map required for training the No-Reference Upscaling Quality Prediction. As shown in Algorithm 1, the No-Reference Upscaling Quality Prediction selects the super-resolution model based on the upscaling quality of the EOSNet little model; thus, the generation of the VMAF map is centered around the EOSNet little model. The generation process involves performing perspective projection on a low-resolution ERP image by adjusting the yaw and pitch at regular intervals to obtain a low-resolution perspective-projected image (LR PP) for each corresponding viewpoint. The LR PP is then upscaled using the EOSNet little model to produce a super-resolved perspective-projected image (SR PP). Similarly, a perspective projection is performed on a high-resolution ERP image to generate a high-resolution perspective-projected image (HR PP) for the same viewpoint. VMAF scores are calculated by using the HR PP as the reference image and the SR PP as the distorted image. This process is repeated for all yaw and pitch angles to construct the VMAF map.

To examine the tendencies in the VMAF map, we generated VMAF maps by shifting the yaw and pitch in 10-degree increments for 24 different 360-degree images. Figure 7a shows the average of the 24 VMAF maps, while Figure 7b presents the VMAF map for a randomly selected image. As shown in the figures, the shape of the VMAF map varies significantly depending on the content of the 360-degree image, differing from any general pattern. This suggests that identifying a general pattern from data is not feasible. Thus, a deep learning-based approach that can extract complex features from the image is necessary to accurately predict the VMAF map.

6.2 VMAF Map Prediction

As illustrated in Figure 8, the VMAF map predictor on the streaming server predicts a VMAF map from a single low-resolution ERP-formatted 360-degree image (LR ERP). Note that high-resolution ERP-formatted 360-degree images (HR ERP), which can be a reference for VMAF, are not available on the streaming server. The efficiency of this map-based approach stems from its ability to predict quality for all potential viewpoints at once, avoiding the computationally expensive process of generating each viewpoint individually.

The VMAF map predictor utilizes DenseNet [28], a type of convolutional neural network (CNN), to extract detailed

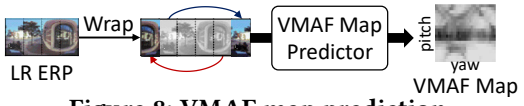


Figure 8: VMAF map prediction.

features from the given LR ERP image. Rather than using the LR ERP image as-is, a wrapped version of the image is used as input to DenseNet. This 'wrapping' method is the process of attaching the leftmost quarter of the LR ERP image to the right end and the rightmost quarter to the left end. The term 'wrapping' describes a specific edge-handling process and is distinct from both general image 'warping' and perspective projection. This adjustment accounts for the fact that, because of the nature of 360-degree images, the left and right edges of the ERP image are connected. In cases where the yaw is close to -180° or 180°, wrapping ensures that features from areas beyond the left and right boundaries are considered. Following the DenseNet layers of the VMAF map predictor, a convolutional layer reduces the number of channels to one to match the VMAF map, followed by an upsampling layer that adjusts the output to match the width and height of the VMAF map. The width corresponds to the number of yaw values, and the height corresponds to the number of pitch values used to generate the VMAF map. The final output layer of the VMAF map predictor applies clamping to ensure that the predicted values always fall within the range of 0 to 100.

Since the EOS SR scheduler decides whether to select the little model based on the $Q_{\text{threshold}}$, it is crucial for the VMAF map prediction to accurately predict scores close to the $Q_{\text{threshold}}$. To this end, we introduced a focal loss function that places emphasis on the area around the $Q_{\text{threshold}}$ when training the VMAF map predictor. The loss function $L(y, \hat{y})$ is defined as follows:

$$\begin{aligned} \text{MSE}(y, \hat{y}) &= \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2, \\ L(y, \hat{y}) &= \text{MSE}(y, \hat{y}) \cdot \left(\exp\left(\frac{-|y - Q_{\text{threshold}}|}{Q_{\text{threshold}}}\right)\right)^2, \end{aligned} \tag{4}$$

where y is the ground-truth value, and $\hat{y}$ is the predicted value. As the ground-truth value gets closer to the $Q_{\text{threshold}}$, the $L(y, \hat{y})$ value increases, making the predictor more sensitive to values near the $Q_{\text{threshold}}$ during training.

6.3 VMAF Score Lookup

The viewing mobile device estimates the VMAF score of a super-resolved viewport by looking up the VMAF map based on the user's current viewing angle. Since the output by the VMAF map predictor is based on yaw and pitch at regular intervals, the VMAF map cannot cover finer yaw and pitch values that fall between these intervals. Therefore, the VMAF score for an arbitrary (yaw, pitch) is obtained through

bilinear interpolation, using the VMAF scores of the four points on the VMAF map that are closest to the given (yaw, pitch) as follows:

$$
\begin{aligned}
Q(x^*, y_1) &= \frac{x_2 - x^*}{x_2 - x_1} \cdot Q(x_1, y_1) + \frac{x^* - x_1}{x_2 - x_1} \cdot Q(x_2, y_1), \\
Q(x^*, y_2) &= \frac{x_2 - x^*}{x_2 - x_1} \cdot Q(x_1, y_2) + \frac{x^* - x_1}{x_2 - x_1} \cdot Q(x_2, y_2), \\
Q(x^*, y^*) &= \frac{y_2 - y^*}{y_2 - y_1} \cdot Q(x^*, y_1) + \frac{y^* - y_1}{y_2 - y_1} \cdot Q(x^*, y_2).
\end{aligned} \tag{5}
$$

Here, $Q(x, y)$ represents the VMAF score for yaw x and pitch y. x^* and y^* are the yaw and pitch values for the current viewing angle, respectively. x_1 is the largest yaw value in the VMAF map among those smaller than x^*, and x_2 is the smallest yaw value among those larger than x^*. Similarly, y_1 is the largest pitch value in the VMAF map among those smaller than y^*, and y_2 is the smallest pitch value among those larger than y^*.

7 Implementation

We implemented the end-to-end EOS system, which consists of a live streaming server and an EOS SR-enabled video player application for Android smartphones. The streaming server uses Real-Time Messaging Protocol (RTMP) through FFmpeg [1] to store low-resolution ODVs received from the streamer as Transport Stream (TS) files. Additionally, the VMAF map predictor running on the streaming server produces VMAF maps from low-resolution ODVs with the PyTorch framework [45]. The streaming server transmits the requested TS files and VMAF maps to mobile devices through a Hypertext Transfer Protocol (HTTP) server program.

The video player application receives ODV live streams from the streaming server, performs super-resolution, and displays the video on the screen. Upon receiving ODV live streams, the video player application decodes the streams using Android's AMediaCodec [3] and performs perspective projection on the decoded ODV frame using OpenGL ES [6]. The perspective-projected frame is then super-resolved using the EOSNet model, which operates through the TensorFlow Lite framework [8], and rendered on the screen. The application adjusts the GPU frequency via the Android system's `sysfs`. The application also includes a background monitoring service to read the GPU frequency, temperature, current, and voltage directly from the Android system's `sysfs`. Note that the application targets rooted Android smartphones because `sysfs` requires root privileges.

8 Evaluation

We evaluated EOSNet and the VMAF map predictor. We also conducted a performance evaluation and a user study under realistic mobile ODV live-streaming scenarios.

8.1 Evaluation Setup

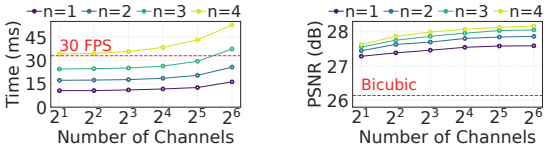
To ensure consistency across experimental conditions, we emulated live streaming using pre-recorded videos. The experiments assumed a network environment described in Section 2.2, where the server and the mobile devices are connected over a Wi-Fi network with a guaranteed 10 Mbps bandwidth. Since public ODV datasets [13, 52] were unsuitable for evaluating EOS due to their limited resolution and length, we curated our dataset: we captured 5-minute ODVs across 16 distinct locations, comprising 12 outdoor and 4 indoor environments, with an Insta360 X3 camera [2]. The ODVs had an overall resolution of 5120×2560 in ERP format with a frame rate of 30 FPS, and were downscaled by a factor of four to 1280×640 for the low-resolution inputs. When applying perspective projection to a given viewpoint, the high-resolution image had a resolution of 1280×720, while the low-resolution image was 320×180. Thus, the on-device super-resolution task was to upscale frames from 320×180 to 1280×720 by a factor of four. The streaming server used in this experiment was equipped with an Intel(R) Xeon(R) Silver 4214 CPU and an NVIDIA GeForce RTX 3090 GPU.

The EOSNet used in the evaluation was trained for 60 epochs on the REDS train dataset, which consists of 320×180 low-resolution and 1280×720 high-resolution images, using PyTorch. For the VMAF map predictor, we selected 8 videos from our ODV dataset, including 6 outdoor and 2 indoor videos, as the training dataset. The predictor was trained with the model for 100 epochs using PyTorch. The remaining videos not included in the training dataset were designated as the test dataset. In this experiment, yaw and pitch of the VMAF map were set at one-degree intervals, and VMAF maps were generated and validated every one second.

We selected two commodity smartphones: a Google Pixel 6 Pro and a Google Pixel 8 Pro. As described in Section 7, the EOS system requires root privileges; therefore, both smartphones used in the experiment were rooted using Magisk [9]. Real ODV streaming experiments were conducted using the video player detailed in Section 7. To minimize the impact of display settings, screen brightness was set to the lowest level in all experiments. When GPU frequency scaling by EOS's policy was not applied, the GPU frequency of the devices was set to the maximum, while the CPU operated under the default governor of the Android operating system. Power consumption was monitored by collecting current and voltage information from the system every 200 ms.

8.2 EOSNet

We conducted benchmark tests on various EOSNet models to evaluate their upscaling quality, inference time, and power consumption. In the experiment, we measured the Peak Signal-to-Noise Ratio (PSNR) using the validation dataset



(a) Inference time on Pixel 6 Pro **(b) PSNR**

Figure 9: EOSNet performance across varying (n, c).

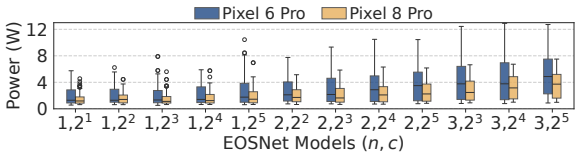
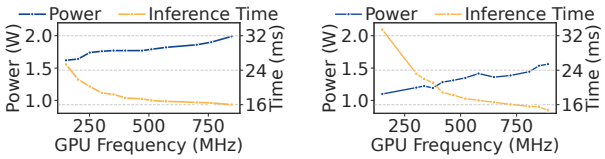


Figure 10: Power consumption.



(a) Pixel 6 Pro **(b) Pixel 8 Pro**

Figure 11: Effect of GPU frequency.

from the REDS dataset. In these benchmark tests, we performed super-resolution tasks at 30 FPS, omitting other tasks such as video rendering. Each model's inference time and power consumption were measured for 1 minute at 30 FPS, resulting in a total of 1,800 measurements.

Figure 9 shows model performance in terms of PSNR and the inference time measured on the Pixel 6 Pro. In the figure, n and c represent the number of dense blocks and the channel size of dense blocks in EOSNet, respectively. As shown in Figure 9a, for the same value of n, models with higher inference times correspond to larger values of c. As shown in Figure 9b, all EOSNet models outperform bicubic interpolation in terms of PSNR, indicating the effectiveness of on-device video super-resolution for enhancing visual quality. Furthermore, the figure shows that simply increasing n or c does not necessarily result in a better model, highlighting the importance of selecting an appropriate combination of n and c. For example, within the region where inference time was under approximately 18 ms, an EOSNet model with $n = 1$ was more effective in terms of PSNR compared to $n = 2$. Specifically, an EOSNet model with $n = 1$ and $c = 2^5$ outperformed a model with $n = 2$ and $c = 2$, making the latter configuration redundant. In this way, the EOSNet models used in the system are carefully selected by filtering out the unnecessary models.

Figure 10 shows the power consumption of the filtered EOSNet models from Figure 9. Note that for both devices, a big model with maximum capability was defined by setting n and c to 3 and 2^5, respectively. Models with n greater than or equal to 4 were excluded from this experiment, as they did not meet EOS's time constraint of 30 FPS. The figure demonstrates that more complex EOSNet models lead to higher power consumption. On the Pixel 6 Pro, the biggest model ($n = 3$, $c = 2^5$) consumed an average power of 5.12 W, while the little model ($n = 1$, $c = 2$) consumed an average power of 1.90 W. On the Pixel 8 Pro, the average power consumption of the biggest model was 3.62 W, compared to 1.44 W for the little model. Replacing the biggest model with the little model resulted in average power reductions of 62.96% and 60.15% for the Pixel 6 Pro and Pixel 8 Pro, respectively.

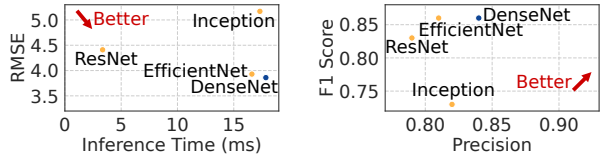
Figure 11 illustrates the impact of GPU frequency on the inference time and power consumption of the EOSNet little model. As shown in the figures, lowering the GPU frequency results in increased inference time and reduced power consumption for both devices. Note that the total energy consumption is directly proportional to the average power, as the system operates at a constant 30 FPS. On the Pixel 6 Pro, even at the minimum GPU frequency, the little model achieved 30 FPS. On the Pixel 8 Pro, however, it was challenging to maintain 30 FPS at the minimum GPU frequency, requiring the selection of a slightly higher frequency as the lowest possible GPU frequency for EOS. These results indicate that, by scaling GPU frequency to the lowest possible level satisfying the 30 FPS requirement, the EOS SR Scheduler achieves average power reductions of 18.59% and 23.08% for the Pixel 6 Pro and Pixel 8 Pro, respectively, compared to running the little model at the maximum GPU frequency.

In conclusion, by controlling the GPU frequency while using the little model, the average power consumption was reduced by 69.85% and 69.35% for the Pixel 6 Pro and Pixel 8 Pro, respectively, compared to the baseline. Note that the results were based on benchmarking DNN inference without actual video playback and rendering, and the power reduction rate may decrease when video playback is included.

8.3 VMAF Map Predictor

We evaluated the performance of the VMAF map predictor used in EOS's No-Reference Upscaling Quality Prediction. For the comparative analysis, we used multiple predictor variants equipped with ResNet-18 [24], EfficientNet-B0 [51], and InceptionV3 [50], along with our DenseNet-based model. All predictors were trained using EOS's focal loss function, and the training data was the same as mentioned in Section 8.1.

Figure 12a shows the inference time and Root Mean Squared Error (RMSE) of the VMAF map predictors. RMSE was measured on a VMAF scale ranging from 0 to 100. As shown in the figure, all predictors produced VMAF maps within 20 ms per frame, enabling 30 FPS operation for low-resolution ODV inputs. These results were obtained with a batch size of 1, and faster inference is possible when using



(a) Latency and RMSE **(b) Precision and F1 score**

Figure 12: Comparison between VMAF map predictors.

larger batch sizes. Among the four predictors, those based on EfficientNet and DenseNet achieved the lowest RMSE.

Figure 12b presents the precision and F1 scores of each predictor. Precision is defined as the proportion of frames predicted to exceed a VMAF score of 80 that actually do exceed that threshold. A higher precision indicates fewer false positives—cases where the predictor incorrectly classifies a frame as meeting the quality threshold—which directly result in QoE degradation. Therefore, among various accuracy metrics, precision is the most critical in this context. As shown in the figure, the DenseNet-based model achieved the highest precision and F1 score among all predictors. When considering all aspects, the DenseNet-based VMAF map predictor is the most suitable.

8.4 EOS SR Scheduler

We conducted a performance evaluation of the EOS system based on real mobile ODV live-streaming scenarios. For comparison, we used the OmniLive-like approach [43], which always selects the most complex model among those meeting the 30 FPS requirement without considering energy efficiency. In the evaluation, both EOS and OmniLive used the EOSNet model for super-resolution.

Live-streaming case study. We first observed various metrics while streaming a live-streaming scenario: model selection, power consumption, inference time, VMAF, GPU frequency, and GPU temperature. For this scenario, we selected one video from the 16 ODVs we collected, excluding those used for training the VMAF map predictor. This video consists of 9,000 frames, running at 30 FPS with a total length of 5 minutes. In this experiment, we defined a viewing scenario to account for the dynamic viewing angle based on user interaction. The yaw and pitch for this viewing scenario throughout the sequence of frames were calculated as follows:

$$\begin{aligned} \text{yaw}(t) &= ((t + 180^\circ) \mod 360^\circ) - 180^\circ, \\ \text{pitch}(t) &= 90^\circ \cdot \sin(\text{yaw}(t)), \end{aligned} \tag{6}$$

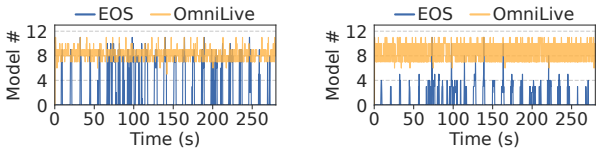
where t refers to the frame number, ranging from 0 to 8,999.

Figure 13 shows the traces of the models selected by each policy. In this graph, a higher model number indicates a more complex EOSNet model, while a model number of 0 represents the selection of the little model. While OmniLive attempted to select the most complex model, EOS tried to select lighter models when possible. Figure 14 shows the traces of the GPU frequencies. Unlike OmniLive, which consistently operated at the maximum GPU frequency, EOS reduced the GPU frequency whenever possible to optimize energy efficiency.

Figure 15 shows the inference time traces for each case. One of the main goals of the EOS SR Scheduler is to ensure that there is no loss in frame rates during on-device video super-resolution. This graph demonstrates that, in all cases, the inference time never exceeds 30 ms. This indicates that the EOS SR Scheduler effectively adheres to the frame rate constraints when selecting EOSNet models and GPU frequency. Figure 16 shows the VMAF traces for each case. Despite EOS selecting lighter models and lower GPU frequencies, there was minimal difference in VMAF scores compared to OmniLive. On the Pixel 6 Pro, the average VMAF scores for the OmniLive and EOS policies were 87.454 and 84.331, respectively. On the Pixel 8 Pro, the scores were 87.935 and 83.606, respectively. This means that compared to OmniLive, EOS resulted in only a small visual quality loss, with average VMAF scores decreasing by 3.56% and 4.92% on the Pixel 6 Pro and the Pixel 8 Pro, respectively.

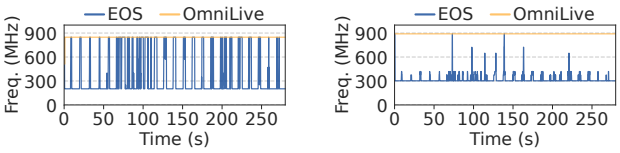
Figure 17 illustrates the power traces, showing that EOS consistently uses less power compared to OmniLive. On the Pixel 6 Pro, EOS recorded an average power consumption of 3.57 W, representing a 37.77% reduction compared to OmniLive's 5.74 W. On the Pixel 8 Pro, EOS demonstrated an average power consumption of 2.27 W, a 50.00% reduction compared to OmniLive's 4.55 W. These results suggest that EOS can achieve significant energy savings with minimal VMAF loss. Additionally, since EOS reduces GPU frequency and utilization, it may also help mitigate overheating, a common issue in smartphones. Figure 18 presents the GPU temperature traces, indicating that EOS results in lower GPU temperatures compared to OmniLive. Specifically, after 5 minutes of live streaming, the GPU temperatures for EOS and OmniLive were 51 °C and 76 °C on the Pixel 6 Pro, and 41 °C and 61 °C on the Pixel 8 Pro, respectively. EOS thus recorded 20 °C to 25 °C lower GPU temperatures. These results demonstrate that EOS's policy is highly effective not only in managing power consumption but also in reducing thermal control in smartphones.

Various live-streaming scenarios. We conducted a comparative analysis of power consumption, GPU temperature, and VMAF between EOS and OmniLive across a broader range of video scenarios. In this experiment, we used 8 ODVs from the 16 we collected, excluding those used for training the VMAF map predictor. To evaluate the effectiveness of EOS SR Scheduler's big/little model selection and GPU frequency scaling techniques, we distinguished between two EOS configurations: EOS with only big/little model selection applied, referred to as EOS (w/o scaling), and EOS with both



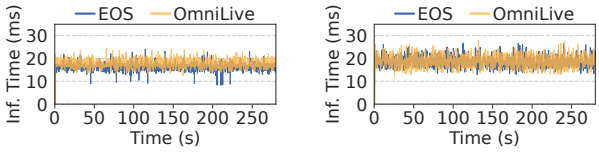
(a) Pixel 6 Pro (b) Pixel 8 Pro

Figure 13: Model selection traces.



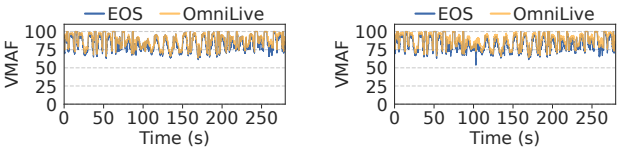
(a) Pixel 6 Pro (b) Pixel 8 Pro

Figure 14: GPU frequency traces.



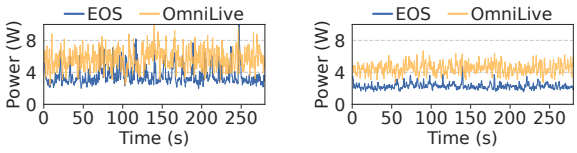
(a) Pixel 6 Pro (b) Pixel 8 Pro

Figure 15: Inference time traces.



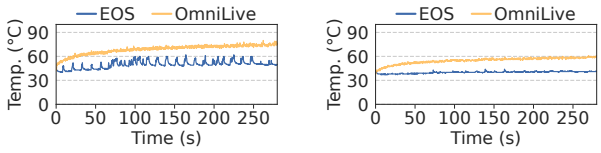
(a) Pixel 6 Pro (b) Pixel 8 Pro

Figure 16: VMAF traces.



(a) Pixel 6 Pro (b) Pixel 8 Pro

Figure 17: Power traces.



(a) Pixel 6 Pro (b) Pixel 8 Pro

Figure 18: GPU temperature traces.

big/little model selection and GPU frequency scaling applied, referred to as (w/ scaling).

Figure 19 illustrates the distribution of power consumption. For the Pixel 6 Pro, EOS (w/o scaling) showed an average power reduction ranging from 29.58% to 37.17%, while EOS (w/ scaling) achieved a reduction between 34.59% and 44.79% compared to OmniLive. On the Pixel 8 Pro, EOS (w/o scaling) reduced average power consumption by 30.52% to 39.82%, and EOS (w/ scaling) demonstrated a reduction from 41.92% to 49.94%. Figure 20 displays the GPU temperature distribution. On the Pixel 6 Pro, EOS (w/o scaling) lowered the average GPU temperature by 11.25 °C to 19.13 °C, and EOS (w/ scaling) further reduced it by 13.64 °C to 23.31 °C compared to OmniLive. On the Pixel 8 Pro, EOS (w/o scaling) showed a decrease in GPU temperature by 8.18 °C to 12.15 °C, while EOS (w/ scaling) reduced the temperature by 12.60 °C to 15.37 °C.

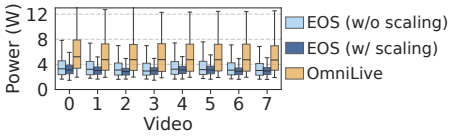
Figure 21 shows the distribution of VMAF scores. On the Pixel 6 Pro, EOS exhibited an average VMAF reduction of 3.27% to 7.28% compared to OmniLive. On the Pixel 8 Pro, EOS showed an average VMAF reduction of 3.76% to 8.18% compared to OmniLive. The cases where a VMAF reduction of 7.28% or 8.18% was observed occurred with Video 7, where OmniLive already had very high average VMAF scores of 92.51 on the Pixel 6 Pro and 93.38 on the Pixel 8 Pro, respectively. Since EOS aims to maintain VMAF scores above 80, VMAF scores significantly higher than 80 make the reduction in percentage terms appear more significant. Overall, EOS's reduction in VMAF remained below 5% in most cases, with the average VMAF score consistently above 80, indicating that the actual reduction in visual quality was minimal.

In conclusion, the results suggest several key findings. EOS effectively optimizes power consumption across various mobile devices and scenarios when performing on-device video super-resolution tasks for mobile ODV live streaming. In addition to its power savings, EOS's operation significantly reduces the heat generated by smartphones. EOS achieves its goal efficiently while maintaining frame rates and minimizing VMAF score loss.

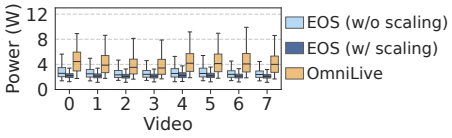
8.5 User Study

We conducted a user study approved by the institutional review board (IRB) to assess the subjective visual quality across three different methods: no super-resolution (No SR), EOS, and OmniLive. A total of 10 participants were recruited for this study: 7 females and 3 males; 5 in their 20s, 3 in their 30s, 1 in their 50s, and 1 in their 60s. For this study, we utilized the same eight video scenarios as in the 'various live-streaming scenarios' evaluation in Section 8.4. For each of the eight video scenarios, participants watched the videos processed by each method and rated the visual quality on a 5-point Likert scale, from 1 (Very Bad) to 5 (Very Good). Figure 22 shows the results of the user study. The mean scores for No SR, EOS, and OmniLive were 1.95, 3.48, and 3.59, respectively. These results suggest that both EOS and OmniLive provide a better visual experience than No SR, while the difference between EOS and OmniLive is minimal.

To statistically validate these observations, we analyzed the significance of the differences between the methods.

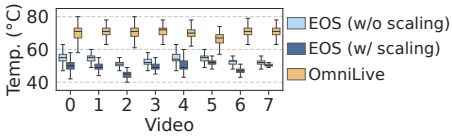


(a) Pixel 6 Pro

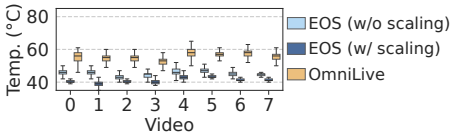


(b) Pixel 8 Pro

Figure 19: Power consumption.

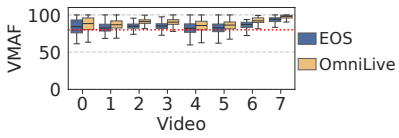


(a) Pixel 6 Pro

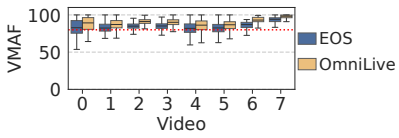


(b) Pixel 8 Pro

Figure 20: GPU temperature.



(a) Pixel 6 Pro



(b) Pixel 8 Pro

Figure 21: VMAF.

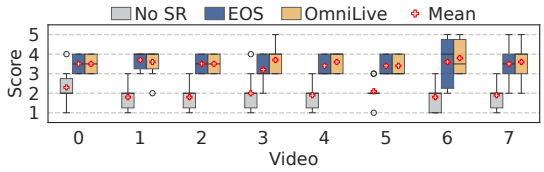


Figure 22: User study results.

Given the small sample size and the ordinal nature of the Likert scale data, which does not satisfy the normality assumption, we performed a Friedman test, followed by a post-hoc analysis using the Wilcoxon signed-rank test with a Bonferroni correction ($p < 0.017$). The Friedman test revealed a statistically significant difference across the three methods ($p < 0.001$). In the post-hoc comparisons, both the No SR vs. EOS and the No SR vs. OmniLive comparisons yielded a p-value of less than 0.001, indicating a clear difference in each case. In contrast, the comparison between EOS and OmniLive yielded a p-value of 0.0389. As the value is greater than the Bonferroni-corrected significance level ($\alpha = 0.017$), no statistically significant difference was detected between the two methods. However, the small sample size limits the statistical power of this study, potentially preventing the detection of a subtle difference between the methods. In summary, the results indicate that EOS not only provides a distinct visual improvement over the No SR baseline but also delivers a subjective visual experience comparable to that of the energy-inefficient OmniLive.

9 Related Work

Super-resolution for video streaming. Super-resolution has been widely adopted to enhance visual quality when high-resolution video delivery is constrained by limited network bandwidth. This includes both traditional 2D video streaming [11, 32, 55, 58] and ODV streaming [20, 23, 25, 26, 49, 56]. However, solutions designed for desktop- or server-class devices are not well-suited for mobile environments due to limited computational capabilities. To address this, some studies have proposed lightweight super-resolution models optimized for mobile execution [29, 40], focusing on minimizing inference latency. Other approaches have aimed to optimize the system with existing neural networks [25, 57, 60, 62, 63]. Nonetheless, these efforts do not comprehensively address the unique challenges in mobile ODV live streaming, namely performing on-device super-resolution under the combined constraints of ODV formats, real-time latency, and limited network bandwidth.

OmniLive [43] is a recent system that introduces on-device super-resolution for mobile ODV live streaming. The system addresses both the limited computational resources of mobile devices and the strict time constraints of live streaming. Specifically, OmniLive employs neural architecture search and multi-exit networks to perform the most complex super-resolution task possible within a given time budget. However, as discussed in Section 3, this approach can lead to unnecessary use of computational resources, resulting in energy inefficiency.

No-reference VMAF. VMAF is known to effectively reflect the perceived visual quality of actual users. However, as a full-reference metric, it requires high-resolution reference videos to function [10, 38]. This requirement makes it inapplicable in scenarios where reference videos are not available, such as in the EOS system. To address this limitation, no-reference VMAF techniques have been proposed [21, 22]. These techniques use deep learning to predict VMAF scores solely from distorted images, without requiring reference data. Unfortunately, existing no-reference VMAF techniques are difficult to use in the EOS system for two reasons. First, since these techniques do not consider super-resolution, using them would require applying super-resolution to every low-resolution image, resulting in additional overhead. Second, these techniques target standard 2D videos, making them difficult to apply to ODVs. As discussed in Section 4.1, ODVs have a vast number of possible viewpoints, making it impractical to apply no-reference VMAF to all possible cases. EOS's No-Reference Upscaling Quality Prediction differs significantly from existing no-reference VMAF approaches in that it predicts VMAF maps for multiple viewpoints at once using only a single low-resolution ERP image.

Energy optimization. Optimizing power consumption or energy usage to extend battery life on smartphones has long been a key problem [16, 18, 19, 30, 33, 36, 37, 39, 44, 47]. In particular, there have been studies on energy optimization for standard 2D video or ODV streaming on mobile devices. These studies have been based on policies for adaptive bitrate streaming [41, 53, 54] or CPU governors of mobile systems [14, 15, 17, 27]. However, since these studies assume streaming without video super-resolution, they are difficult to be used in systems where video super-resolution is employed. Meanwhile, NEMO [57] proposed a technique to optimize the time and energy efficiency of on-device video super-resolution for Video-on-Demand (VoD) streaming. However, as NEMO targets traditional 2D videos, its approach is not feasible for ODVs, where dynamically changing viewpoints limit the ability to reuse super-resolution results. Furthermore, NEMO requires a lengthy analysis of video content stored on the server, making it unsuitable for live streaming.

10 Discussion

Generality and QoE Metrics. Unlike traditional formula-based objective metrics such as PSNR and Structural Similarity Index Measure (SSIM), VMAF is an advanced approach that integrates subjective video quality ratings derived from user studies [10, 38]. Moreover, VMAF internally leverages sophisticated metrics that outperform traditional ones, including Visual Information Fidelity (VIF) [48], which models the human visual system, and the Detail Loss Metric (DLM) [35], which captures the loss of detail. This suggests that a VMAF score of 80 serves as a robust benchmark; thus, we believe EOS can generalize effectively across diverse scenarios beyond the specific setup used in Section 8.

Although recent studies have explored new methods [42, 46] and proposed deep learning-based metrics such as LPIPS [64], the EOS system is extensible to other metrics beyond VMAF. For instance, a similar training methodology could be employed to predict a QoE map for a different metric in place of the VMAF map. The EOS SR Scheduler could then operate using the same policy, simply adapting to the new quality standard.

Deployment on other mobile devices. As discussed in Sections 7 and 8.1, the implementation and evaluation of EOS require low-level system access to control GPU frequency and monitor power consumption, which necessitate root privileges on most commercial smartphones. While this presents a deployment challenge, device manufacturers, driven by the need for energy efficiency, could integrate EOS directly at the system level or provide interfaces for GPU frequency control. Even without GPU frequency scaling, the big/little model selection scheme, which does not require root access, still offers substantial energy optimization. Furthermore, we argue that EOS's approach is applicable to Android-based Head-Mounted Displays (HMDs), such as the Meta Quest series [4], as they utilize similar processor architectures to smartphones and are subject to significant thermal and power constraints.

Applicability to other streaming types. While the current EOS system focuses on mobile ODV live streaming, the core principles of EOS are readily applicable to other streaming types. For instance, in the case of live streaming for standard 2D videos, EOS's No-Reference Upscaling Quality Prediction can be simplified to predict a single VMAF score for the frame instead of an entire VMAF map. Additionally, even for mobile ODV live streaming utilizing viewport prediction [61], EOS's approach to predict a VMAF map for the entire viewing angle at once is still beneficial. Since live viewport prediction is inherently prone to errors, EOS's approach provides a robust fallback for cases where the user's gaze deviates from the predicted area.

11 Conclusion and Future Work

To the best of our knowledge, EOS is the first attempt to provide energy- and QoE-aware on-device video super-resolution for mobile ODV live streaming. Through the No-Reference Upscaling Quality Prediction, EOS effectively predicts the VMAF scores for various viewpoints using only low-resolution ODVs. Based on the predicted VMAF scores, EOS SR successfully optimizes average power consumption while minimizing losses in both visual quality and frame rates during ODV playback on mobile devices.

For future work, we plan to extend EOS in two primary directions. First, we will adapt the system to support adaptive bitrate (ABR) streaming by managing a separate VMAF map predictor for each resolution, allowing EOS to respond to dynamic network conditions. Second, while the current two-tier (big/little) model was sufficient for contemporary high-end smartphones, we envision a multi-level hierarchy of common models (e.g., little, medium, big) for future devices with more powerful GPUs, which would enable more fine-grained energy policies.

Acknowledgments

This work was partly supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (No. RS-2024-00412632, 34%), the NRF grant funded by the Korea government (MSIT) (No. RS-2024-00344323, 33%), and Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. RS-2018-II180532, 33%).

References

[1] [n. d.]. FFmpeg. https://ffmpeg.org/

[2] [n. d.]. Insta360 X3. https://www.insta360.com/product/insta360-x3

[3] [n. d.]. MediaCodec. https://developer.android.com/reference/android/media/MediaCodec

[4] [n. d.]. Meta Quest. https://www.meta.com/quest/

[5] [n. d.]. Netflix/VMAF. https://github.com/Netflix/vmaf

[6] [n. d.]. OpenGL. https://www.opengl.org/

[7] [n. d.]. Speedtest. https://www.speedtest.net/global-index

[8] [n. d.]. TensorFlow Lite. https://www.tensorflow.org/lite

[9] [n. d.]. topjohnwu/Magisk. https://github.com/topjohnwu/Magisk

[10] [n. d.]. VMAF: The Journey Continues. https://netflixtechblog.com/vmaf-the-journey-continues-44b51ee9ed12

[11] Hadi Amirpour, Mohammad Ghanbari, and Christian Timmerer. 2022. DeepStream: Video Streaming Enhancements using Compressed Deep Neural Networks. *IEEE Transactions on Circuits and Systems for Video Technology* (2022). doi:10.1109/TCSVT.2022.3229079

[12] Jose Caballero, Christian Ledig, Andrew Aitken, Alejandro Acosta, Johannes Totz, Zehan Wang, and Wenzhe Shi. 2017. Real-time video super-resolution with spatio-temporal networks and motion compensation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2848–2857. doi:10.1109/CVPR.2017.304

[13] Mingdeng Cao, Chong Mou, Fanghua Yu, Xintao Wang, Yinqiang Zheng, Jian Zhang, Chao Dong, Gen Li, Ying Shan, Radu Timofte, Xiaopeng Sun, Weiqi Li, Zhenyu Zhang, Xuhan Sheng, Bin Chen, Haoyu Ma, Ming Cheng, Shijie Zhao, Wanwan Cui, Tianyu Xu, Chunyang Li, Long Bao, Heng Sun, Huaibo Huang, Xiaoqiang Zhou, Yuang Ai, Ran He, Renlong Wu, Yi Yang, Zhilu Zhang, Shuohao Zhang, Junyi Li, Yunjin Chen, Dongwei Ren, Wangmeng Zuo, Qian Wang, Hao-Hsiang Yang, Yi-Chung Chen, Zhi-Kai Huang, Wei-Ting Chen, Yuan-Chun Chiang, Hua-En Chang, I-Hsiang Chen, Chia-Hsuan Hsieh, Sy-Yen Kuo, Zebin Zhang, Jiaqi Zhang, Yuhui Wang, Shuhao Cui, Junshi Huang, Li Zhu, Shuman Tian, Wei Yu, and Bingchun Luo. 2023. NTIRE 2023 Challenge on 360deg Omnidirectional Image and Video Super-Resolution: Datasets, Methods and Results. In *2023 IEEE/CVF Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 1731–1745.

[14] Xianda Chen and Guohong Cao. 2022. Energy-Efficient and QoE-Aware 360-Degree Video Streaming on Mobile Devices. In *IEEE 42nd International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 994–1004. doi:10.1109/ICDCS54860.2022.00100

[15] Xianda Chen and Guohong Cao. 2023. Energy-Efficient 360-Degree Video Streaming on Multicore-Based Mobile Devices. In *IEEE INFOCOM 2023 - IEEE Conference on Computer Communications*, Vol. 2023-May. IEEE. doi:10.1109/INFOCOM53939.2023.10228863

[16] Xiaomeng Chen, Ning Ding, Abhilash Jindal, Y. Charlie Hu, Maruti Gupta, and Rath Vannithamby. 2015. Smartphone Energy Drain in the Wild. *ACM SIGMETRICS Performance Evaluation Review* 43, 1 (6 2015), 151–164. doi:10.1145/2796314.2745875

[17] Xianda Chen, Tianxiang Tan, and Guohong Cao. 2024. Macrotile: Toward QoE-Aware and Energy-Efficient 360-Degree Video Streaming. *IEEE Transactions on Mobile Computing* 23, 2 (2 2024), 1112–1126. doi:10.1109/TMC.2022.3233022

[18] Yonghun Choi, Seonghoon Park, and Hojung Cha. 2019. Graphics-aware power governing for mobile devices. In *Proceedings of the 17th Annual International Conference on Mobile Systems, Applications, and Services (MobiSys '19)*. ACM, 469–481. doi:10.1145/3307334.3326075

[19] Yonghun Choi, Seonghoon Park, and Hojung Cha. 2019. Optimizing Energy Efficiency of Browsers in Energy-Aware Scheduling-enabled Mobile Devices. In *Proceedings of the 25th Annual International Conference on Mobile Computing and Networking (MobiCom '19)*. ACM, 1–16. doi:10.1145/3300061.3345449

[20] Mallesham Dasari, Arani Bhattacharya, Santiago Vargas, Pranjal Sahu, Aruna Balasubramanian, and Samir R. Das. 2020. Streaming 360-Degree Videos Using Super-Resolution. In *IEEE INFOCOM 2020 - IEEE Conference on Computer Communications*. IEEE, 1977–1986. doi:10.1109/INFOCOM41043.2020.9155477

[21] Axel De Decker, Jan De Cock, Peter Lambert, and Glenn Van Wallendael. 2024. No-Reference VMAF: A Deep Neural Network-Based Approach to Blind Video Quality Assessment. *IEEE Transactions on Broadcasting* (2024). doi:10.1109/TBC.2024.3399479

[22] Lina Du, Shuo Yang, Li Zhuo, Hui Zhang, Jing Zhang, and Jiafeng Li. 2022. Quality of Experience Evaluation Model with No-Reference VMAF Metric and Deep Spatio-temporal Features of Video. *Sensing and Imaging* 23, 1 (12 2022), 1–22. doi:10.1007/S11220-022-00386-2/TABLES/9

[23] Beizhang Guo, Juntao Bao, Baili Chai, Di Wu, and Miao Hu. 2024. Lumos: Optimizing Live 360-degree Video Upstreaming via Spatial-Temporal Integrated Neural Enhancement. In *Proceedings of the 32nd ACM International Conference on Multimedia (MM '24)*. ACM, 7210–7219. doi:10.1145/3664647.3681305

[24] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 770–778. doi:10.1109/CVPR.2016.90

[25] Siyuan Hong, Ruiqi Wang, and Guohong Cao. 2025. Adaptive 360-Degree Video Streaming with Super-Resolution and Interpolation. In *2025 IEEE Conference Virtual Reality and 3D User Interfaces (VR)*. 34–43. doi:10.1109/VR59515.2025.00028

[26] Biao Hou, Song Yang, Fan Li, Liehuang Zhu, Xu Chen, Yu Wang, and Xiaoming Fu. 2024. NOVA: Neural-Optimized Viewport Adaptive 360-Degree Video Streaming at the Edge. *IEEE Transactions on Services Computing* 17, 6 (2024), 4027–4040. doi:10.1109/TSC.2024.3451237

[27] Wenjie Hu and Guohong Cao. 2017. Energy-Aware CPU Frequency Scaling for Mobile Video Streaming. In *IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 2314–2321. doi:10.1109/ICDCS.2017.74

[28] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q. Weinberger. 2017. Densely connected convolutional networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2261–2269. doi:10.1109/CVPR.2017.243

[29] Andrey Ignatov, Andres Romero, Heewon Kim, Radu Timofte, Chiu Man Ho, Zibo Meng, Kyoung Mu Lee, Yuxiang Chen, Yutong Wang, Zeyu Long, Chenhao Wang, Yifei Chen, Boshen Xu, Shuhang Gu, Lixin Duan, Wen Li, Wang Bofei, Zhang Diankai, Zheng Chengjian, Liu Shaoli, Gao Si, Zhang Xiaofeng, Lu Kaidi, Xu Tianyu, Zheng Hui, Xinbo Gao, Xiumei Wang, Jiaming Guo, Xueyi Zhou, Hao Jia, and Youliang Yan. 2021. Real-time video super-resolution on smartphones with deep learning, mobile AI 2021 challenge: Report. In *2021 IEEE/CVF Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. IEEE, 2535–2544. doi:10.1109/CVPRW53098.2021.00287

[30] Abhilash Jinda and Y. Charlie Hu. 2018. Differential Energy Profiling: Energy Optimization via Diffing Similar Apps. In *Proceedings of the 13th USENIX conference on Operating Systems Design and Implementation (OSDI '18)*. USENIX, Carlsbad, CA.

[31] Chanyoung Jung, Jeho Lee, Gunjoong Kim, Jiwon Kim, Seonghoon Park, and Hojung Cha. 2025. ARIA: Optimizing Vision Foundation Model Inference on Heterogeneous Mobile Processors for Augmented Reality. In *Proceedings of the 23rd Annual International Conference on Mobile Systems, Applications and Services (MobiSys '25)*. ACM, New York, NY, USA. doi:10.1145/3711875.3729161

[32] Jaehong Kim, Youngmok Jung, Hyunho Yeo, Juncheol Ye, and Dongsu Han. 2020. Neural-Enhanced Live Streaming: Improving Live Video Ingest via Online Learning. In *Proceedings of the 2020 Annual Conference of the ACM Special Interest Group on Data Communication on the Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM '20)*. ACM, 107–125. doi:10.1145/3387514.3405856

[33] Seyeon Kim, Kyungmin Bin, Sangtae Ha, Kyunghan Lee, and Song Chong. 2021. ZTT: Learning-based DVFS with zero thermal throttling for mobile devices. In *Proceedings of the 19th Annual International Conference on Mobile Systems, Applications, and Services (MobiSys '21)*. ACM, 41–53. doi:10.1145/3458864.3468161

[34] Jeho Lee, Chanyoung Jung, Jiwon Kim, and Hojung Cha. 2024. Panopticus: Omnidirectional 3D Object Detection on Resource-constrained Edge Devices. In *Proceedings of the 30th Annual International Conference on Mobile Computing and Networking (MobiCom '24)*. ACM, New York, NY, USA, 1207–1221. doi:10.1145/3636534.3690688

[35] Songnan Li, Fan Zhang, Lin Ma, and King Ngi Ngan. 2011. Image Quality Assessment by Separately Evaluating Detail Losses and Additive Impairments. *IEEE Transactions on Multimedia* 13, 5 (2011), 935–949. doi:10.1109/TMM.2011.2152382

[36] Xueliang Li, Junyang Chen, Yepang Liu, Kaishun Wu, and John P. Gallagher. 2023. Combatting Energy Issues for Mobile Applications. *ACM Transactions on Software Engineering and Methodology* 32, 1 (2 2023). doi:10.1145/3527851

[37] Xueliang Li, Shicong Hong, Junyang Chen, Junkai Ji, Chengwen Luo, Guihai Yan, Zhibin Yu, and Jianqiang Li. 2024. Satisfying Energy-Efficiency Constraints for Mobile Systems. *IEEE Transactions on Mobile Computing* (2024). doi:10.1109/TMC.2024.3447026

[38] Zhi Li, Anne Aaron, Ioannis Katsavounidis, Anush Moorthy, and Megha Manohara. 2016. Toward A Practical Perceptual Video Quality Metric. https://netflixtechblog.com/toward-a-practical-perceptual-video-quality-metric-653f208b9652

[39] Chengdong Lin, Kun Wang, Zhenjiang Li, and Yu Pu. 2023. A Workload-Aware DVFS Robust to Concurrent Tasks for Mobile Devices. In *Proceedings of the 29th Annual International Conference on Mobile Computing and Networking (MobiCom '23)*. ACM, 270–285. doi:10.1145/3570361.3592524

[40] Shaoli Liu, Chengjian Zheng, Kaidi Lu, Si Gao, Ning Wang, Bofei Wang, Diankai Zhang, Xiaofeng Zhang, and Tianyu Xu. 2021. EVSRNet: Efficient video super-resolution with neural architecture search. In *2021 IEEE/CVF Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. IEEE, 2480–2485. doi:10.1109/CVPRW53098.2021.00281

[41] Jiayi Meng, Qiang Xu, and Y Charlie Hu. 2021. Proactive Energy-Aware Adaptive Video Streaming on Mobile Devices. In *2021 USENIX Annual Technical Conference (USENIX ATC '21)*. USENIX, 303–316.

[42] Xiongkuo Min, Huiyu Duan, Wei Sun, Yucheng Zhu, and Guangtao Zhai. 2024. *Perceptual Video Quality Assessment: A Survey*. Technical Report. arXiv. https://arxiv.org/abs/2402.03413v1

[43] Seonghoon Park, Yeonwoo Cho, Hyungchol Jun, Jeho Lee, and Hojung Cha. 2023. OmniLive: Super-Resolution Enhanced 360° Video Live Streaming for Mobile Devices. In *Proceedings of the 21st Annual International Conference on Mobile Systems, Applications and Services (MobiSys '23)*. ACM, 261–274. doi:10.1145/3581791.3596851

[44] Seonghoon Park, Jiwon Kim, Jeho Lee, and Hojung Cha. 2025. Ember: Task Wakeup Sequence–Based Energy Optimization for Mobile Web Browsing. *ACM Trans. Embed. Comput. Syst.* (2025). doi:10.1145/3757918

[45] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury Google, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf Xamla, Edward Yang, Zach Devito, Martin Raison Nabla, Alykhan Tejani, Sasank Chilamkurthy, Qure Ai, Benoit Steiner, Lu Fang Facebook, Junjie Bai Facebook, and Soumith Chintala. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems (NeurIPS '19)*. doi:10.5555/3454287.3455008

[46] Avinab Saha, Sai Karthikey Pentapati, Zaixi Shang, Ramit Pahwa, Bowen Chen, Hakan Emre Gedik, Sandeep Mishra, and Alan C. Bovik. 2023. Perceptual video quality assessment: the journey continues! *Frontiers in Signal Processing* 3 (6 2023), 1193523. doi:10.3389/FRSIP.2023.1193523

[47] Qianlong Sang, Jinqi Yan, Rui Xie, Chuang Hu, Kun Suo, and Dazhao Cheng. 2024. QoS-Aware Power Management via Scheduling and Governing Co-Optimization on Mobile Devices. *IEEE Transactions on Mobile Computing* 23, 12 (2024), 13654–13669. doi:10.1109/TMC.2024.3438267

[48] H R Sheikh and A C Bovik. 2006. Image information and visual quality. *IEEE Transactions on Image Processing* 15, 2 (2006), 430–444. doi:10.1109/TIP.2005.859378

[49] Jianxin Shi, China Lingjun Pu, China Xinjing Yuan, China Qianyun Gong, China Jingdong Xu, Lingjun Pu, Xinjing Yuan, Qianyun Gong, and Jingdong Xu. 2022. Sophon: Super-Resolution Enhanced 360° Video Streaming with Visual Saliency-aware Prefetch. In *Proceedings of the 30th ACM International Conference on Multimedia (MM '22)*. ACM, 3124–3133. doi:10.1145/3503161.3547750

[50] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the Inception Architecture for Computer Vision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2818–2826.

[51] Mingxing Tan and Quoc Le. 2019. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. In *Proceedings of the 36th International Conference on Machine Learning (ICML '19)*. 6105–6114.

[52] Ahmed Telili, Ibrahim Farhat, Wassim Hamidouche, and Hadi Amirpour. 2024. ODVISTA: An Omnidirectional Video Dataset for Super-Resolution and Quality Enhancement Tasks. In *2024 IEEE International Conference on Image Processing (ICIP)*. 131–136. doi:10.1109/ICIP51287.2024.10647612

[53] Bekir Oguzhan Turkkan, Ting Dai, Adithya Raman, Tevfik Kosar, Changyou Chen, Muhammed Bulut, Jaroslav Zola, and Daby Sow. 2024. GreenABR+: Generalized Energy-Aware Adaptive Bitrate Streaming. *ACM Transactions on Multimedia Computing, Communications and Applications* 20, 9 (8 2024), 1–24. doi:10.1145/3649898

[54] Bekir Oguzhan Turkkan, Ting Dai, Adithya Raman, Tevfik Kosar, Changyou Chen, Muhammed Fatih Bulut, Jaroslaw Zola, and Daby Sow. 2022. GreenABR: Energy-Aware Adaptive Bitrate Streaming with Deep Reinforcement Learning. In *Proceedings of the 13th ACM Multimedia Systems Conference (MMSys '22)*, Vol. 22. ACM, 150–163. doi:10.1145/3524273.3528188

[55] Lingdong Wang, Simran Singh, Jacob Chakareski, Mohammad Hajiesmaili, and Ramesh K Sitaraman. 2024. BONES: Near-Optimal Neural-Enhanced Video Streaming. *Proc. ACM Meas. Anal. Comput. Syst.* 8, 2 (2024). doi:10.1145/3656014

[56] Zhengguan Wu, Jingwei Liao, Anh Nguyen, Feng Lin, and Zhisheng Yan. 2025. Orbis: Redesigning Neural-enhanced Video Streaming for Live Immersive Viewing. In *Proceedings of the 23rd ACM Conference on Embedded Networked Sensor Systems (SenSys '25)*. ACM, 371–384. doi:10.1145/3715014.3722088

[57] Hyunho Yeo, Chan Ju Chong, Youngmok Jung, Juncheol Ye, and Dongsu Han. 2020. NEMO: Enabling Neural-enhanced Video Streaming on Commodity Mobile Devices. In *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking (MobiCom '20)*. ACM, 1–14. doi:10.1145/3372224.3419185

[58] Hyunho Yeo, Youngmok Jung, Jaehong Kim, Jinwoo Shin, and Dongsu Han. 2018. Neural Adaptive Content-aware Internet Video Delivery. In *Proceedings of the 13th USENIX conference on Operating Systems Design and Implementation (OSDI '18)*. USENIX, Carlsbad, CA, 645–661.

[59] Hyunho Yeo, Hwijoon Lim, Jaehong Kim, Youngmok Jung, Juncheol Ye, and Dongsu Han. 2022. NeuroScaler: Neural video enhancement at scale. In *Proceedings of the ACM SIGCOMM 2022 Conference (SIGCOMM '22)*. ACM, 795–811. doi:10.1145/3544216.3544218

[60] Qian Yu, Qing Li, Rui He, Gareth Tyson, Wanxin Shi, Jianhui Lv, Zhenhui Yuan, Peng Zhang, Yulong Lan, and Zhicheng Li. 2023. BiSR: Bidirectionally Optimized Super-Resolution for Mobile Video Streaming. In *Proceedings of the ACM Web Conference 2023 (WWW '23)*. ACM, 3121–3131. doi:10.1145/3543507.3583519

[61] Lei Zhang, Peng Chen, Cong Zhang, Cheng Pan, Tao Long, Weizhen Xu, Laizhong Cui, and Jiangchuan Liu. 2025. Optimizing Mobile-Friendly Viewport Prediction for Live 360-Degree Video Streaming. *IEEE Transactions on Mobile Computing* (2025), 1–14. doi:10.1109/TMC.2025.3571186

[62] Lei Zhang, Haotian Guo, Yanjie Dong, Fangxin Wang, Laizhong Cui, and Victor C M Leung. 2023. Collaborative Streaming and Super Resolution Adaptation for Mobile Immersive Videos. In *IEEE INFOCOM 2023 - IEEE Conference on Computer Communications*. IEEE, 1–10. doi:10.1109/INFOCOM53939.2023.10228906

[63] Lei Zhang, Haobin Zhou, Haiyang Wang, and Laizhong Cui. 2024. TBSR: Tile-Based 360° Video Streaming with Super-Resolution on Commodity Mobile Devices. In *IEEE INFOCOM 2024 - IEEE Conference on Computer Communications*. IEEE, 1–10. doi:10.1109/INFOCOM52122.2024.10621078

[64] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. 2018. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 586–595. doi:10.1109/CVPR.2018.00068