DoMo: Rethinking Downscaling for Mobile Neural-enhanced Video Streaming

Zhui Zhu[*], Xu Wang[†✉], Jingao Xu[‡], Weichen Zhang[†], Yankun Yuan[§], Lin Wang[§] Fan Dang[¶], Yunhao Liu[*]
[*]Department of Automation and BNRist, Tsinghua University, [†]Global Innovation Exchange, Tsinghua University
[‡]Computer Science Department, Carnegie Mellon University
[§]School of Information Science and Engineering, Yanshan University
[§]School of Software Engineering, Beijing Jiaotong University
{z-zhu22,weic_zhang23}@mails.tsinghua.edu.cn, {yyk13964371891,xujingao13,wangllinn}@gmail.com
{xu_wang, yunhao}@tsinghua.edu.cn, dangfan@bjtu.edu.cn

IEEE INFOCOM 2025 - IEEE Conference on Computer Communications | 979-8-3315-4305-1/25/$31.00 ©2025 IEEE | DOI: 10.1109/INFOCOM55648.2025.11044486

***Abstract*—With the prevalence of 4G/5G infrastructure and mobile devices, mobile video streaming has become an ubiquitous element of daily life. Nevertheless, the online delivery of high-resolution videos, such as 2K and 4K formats, encounters significant challenges due to bandwidth limitations and network fluctuations. Existing neural-enhanced video streaming systems primarily struggle with two issues: the difficulty of recovering intra-frame high-frequency content and reusing the inter-frame content correlation. Addressing these challenges, this paper introduces a novel approach, designated as DoMo, which reconsiders the potential of mobile-side video super-resolution (SR) from a cloud perspective. We implement DoMo for the VP9 codec and test on real on-demand streaming media videos. Empirical results indicate that DoMo not only surpasses current state-of-the-art neural-enhanced solutions by achieving a 3.32 - 4.54 dB improvement in the peak signal-to-noise ratio (PSNR), but also outperforms traditional non-SR decoding methods by 6.80 - 8.89 dB.**

***Index Terms*—Video Streaming, Artificial Intelligence, Mobile Computing**

I. Introduction

The ubiquity of the 4G/5G infrastructure and mobile devices has made mobile video streaming a fundamental aspect of daily life [1]. Meanwhile, higher resolution video streams (*e.g.*, 4K), are aligning with consumer demands for enhanced video quality and are emerging as a significant trend. This market is predicted to reach a valuation of $1 trillion dollars in the coming years, with mobile video streaming serving as a crucial driver of this market expansion [2], [3]. However, delivering those videos online poses challenges due to bandwidth constraints and network instability [4], [5]. Although efforts have been made to maximize user QoE with bitrate adaptation to manage network fluctuations [6]–[10], they cannot consistently ensure high QoE in diverse visual content and network conditions.

As an alternative, tackling the issue from the perspective of video rather than network, neural enhancement techniques - specifically video super-resolution (SR) - have also hit the mainstream [11]–[16]. Current practice improves video quality by reconstructing high-quality videos from low-resolution (LR) streams acquired on the mobile side, mitigating the effects of poor network conditions. A typical neural-enhanced system comprises two main components: *cloud-side video downscaling* and *mobile-side video SR*. The cloud down-scales high-resolution (HR) videos to lower ones (*e.g.*, 270p or 540p) according to network conditions and re-encodes them into chunks for streaming. Upon receiving them, the mobile employs an SR neural network to recover the video quality.

Albeit inspiring, our experiments with 4K YouTube video data reveal that current efforts still fall short in enhancing user QoE. The underlying issue is that existing solutions concentrate on designing and optimizing mobile-side SR neural networks, but overlook *how the cloud should downscale videos to better suit mobile-side SR video reconstruction*. The absence of coordinated design between cloud- and mobile-side frameworks results in two primary challenges:

- **Intra-frame high-frequency content is hard to recover.** The cloud downscales video frames for transmission over networks with limited bandwidth. Current practice commonly employs basic downscaling filters, such as bilinear and bicubic [17]. However, these methods often result in the loss of high-frequency information, which includes critical details such as fine textures and sharp edges, making SR reconstruction challenging and impacting user QoE (§II-B1).
- **Inter-frame content correlation is difficult to reuse.** To enhance the SR efficiency on mobile devices, recent works [13] apply neural network inference selectively to anchor frames and propagate these results to non-anchor frames by interpolating the inter-frame residuals (§II-A). However, these residuals from acquired video streams represent differences between downscaled, LR (*e.g.*, 540p) frames and fail to capture the accurate differences between HR (*e.g.*, 4k) frames after SR upscaling. Consequently, this interpolation process accumulates errors over time, which impedes the effective use of inter-frame content correlations for reconstructing non-anchor frames (§II-B2).

Remark: In summary, the quality of video reconstructed on the mobile through SR is fundamentally linked to the quality of the video initially downscaled by the cloud. Enhancing system performance requires simultaneous optimization of both *cloud-side downscaling* and *mobile-side SR* modules. Specifically, crucial aspects are (i) preserving high-frequency intra-frame

[✉]Xu Wang is the corresponding author.

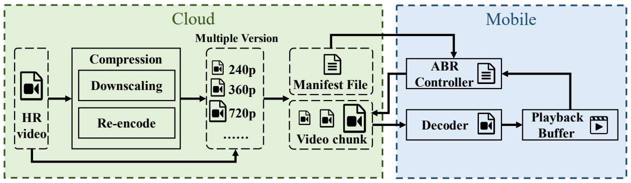


Fig. 1: Video on-demand Delivery System

details during downscaling and (ii) designing video streams that support SR-compatible inter-frame content correlations.

Our Work: We translate the above insights into a practical system and present DoMo, the first work to rethink and unleash the potential of mobile-side video SR from the cloud's perspective. DoMo further enhances video quality through the cloud-mobile co-design.

• On the cloud front, our design incorporates three modules: (i) for intra-frame SR, we design an *Efficient Invertible Neural Network* to encode high-frequency information, replacing simpler filters like Bicubic that typically generate LR videos (§III-A); (ii) for inter-frame SR, a *Reuse Aware Neural Network Training* framework is developed to identify and mitigate video quality loss during frame interpolation in the upscaling process, thereby improving video quality (§III-B); and (iii) a *Video Frames Joint Selector* is further designed to reduce overall errors in reusing decoded frames (§III-D).

• On the mobile front, we have developed a *Ref-based Neural Decoder*, an enhancement of the existing SR-integrated decoder, to sync with advances on the cloud side, effectively addressing intra-frame and inter-frame challenges and enabling superior video frame reconstruction (§III-C).

We have expanded libvpx [18] to fully implement DoMo for VP9 [19] and tested it on real on-demand streaming videos. Our experiments demonstrate that, at equivalent bitrates and throughput, DoMo significantly improves video quality (in PSNR), achieving gains of 3.32 - 4.54 dB over the state-of-the-art solution (*i.e.*, Nemo [13]) and 6.80 - 8.89 dB over conventional decoding methods. Furthermore, under identical video quality and bandwidth conditions, DoMo reduces energy consumption by 45.7% compared to Nemo. Furthermore, when tested with real network data and a traditional adaptive bitrate streaming (ABR) algorithm, our system improves the average QoE by 29.6%.

In summary, this paper makes the following contributions:

- We conduct a systematic analysis of the key limitations in existing neural-enhanced video delivery systems and, on this basis, introduce a novel perspective to optimize mobile-side video SR from the cloud's viewpoint.
- We design a closed-loop neural-enhanced on-demand video streaming system through cloud-mobile co-design.
- Extensive evaluations demonstrate that our system achieves improved video quality, QoE, and power consumption performance.

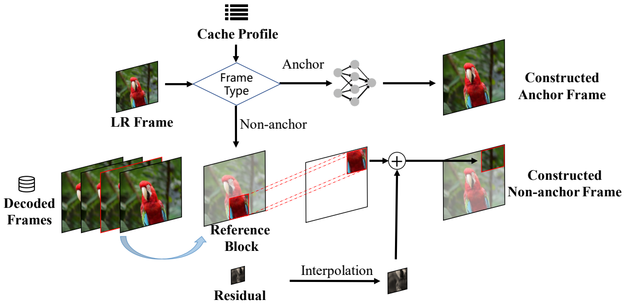


Fig. 2: Nemo [13] applied neural network inference to anchor frames and propagated these results to non-anchor frame.

II. Background and Motivation

A. Primer on Neural-enhanced Video On-demand Delivery

Recent advances in SR integrated video streaming bring a new chance for QoE enhancement using mobile-side computation independent of the network bandwidth. In this scheme, videos in the cloud are compressed into multiple versions with various bitrates and resolutions by downscaling and re-encoding. When some mobile send a request for a video, the mobile downloads an LR version video and applies SR on the video frames to reconstruct the HR visuals. To make the process real-time and energy-efficient, pioneer studies [13], [16] avoid applying SR on every frame but transfer the pixels of cached HR frames to other frames.

Among them, the state-of-the-art work Nemo [13] implements the transfer by using residuals and motion vectors from the video codec to compensate for the temporal difference. The key method is illustrated in Fig. 2. Nemo divides video frames into anchor frames and non-anchor frames. Anchor frames undergo up-scaling using an SR model to enhance their detail and clarity. Non-anchor frames are upscaled by reusing the HR result of dependent anchor frames. More precisely, each HR block b_j^h of a non-anchor frame j is matched with the dependent HR block b_i^h in the previous anchor frame i. The residual r_{ij}^h between b_j^h and b_i^h is estimated by interpolation upscaling on the residual r_{ij}^l between the LR frames i and j. This can be formally expressed as follows:

$$b_j^h = b_i^h + \mathit{UPSCALING}(r_{ij}^l). \tag{1}$$

Although practical, existing solutions focus on designing and optimizing mobile-side SR, and overlook how the cloud downscales LR video, which makes it hard to recover high-frequency content and reuse content correlation.

B. The Challenges of Previous mobile-side SR Research

Mobile-side SR enables real-time decoding of mobile video streaming using neural networks, but it presents certain challenges. The quality of the decoded video hinges on two critical aspects: (1) high-frequency content is missing and (2) the loss of quality due to residual interpolation when reusing decoded frames. We analyze the state-of-the-art system with 4K YouTube videos, and our experiments reveal limitations in

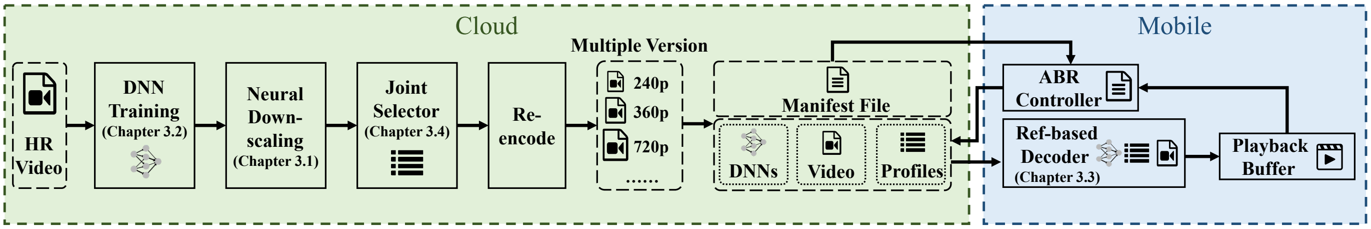
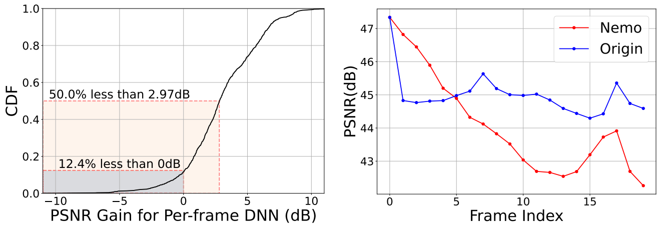


Fig. 3: System Overview



(a) Per-frame DNN Result

(b) Quality Decline of SR-integrated codec

Fig. 4: Experiments on video data show the limitation of Nemo [13].

both areas, resulting in a notable degradation of video quality, additional power consumption, and bandwidth consumption.

1) From Intra Frame's Perspective: Firstly, we find that the SR network exhibits unstable performance across different chunks of videos (Fig. 4a). We utilized neural networks within the NEMO [13] to upscale 540p videos to 2160p, and subsequently compared the results with the original 540p videos. In 12.4% of the chunks, the average quality decoded by per-frame DNNs is lower than that decoded directly through interpolation of the original VP9 video. And in 50% of the chunks, the quality gain is less than 2.97dB.

SR algorithms define the highest achievable video quality in SR-based video streaming systems. However, the result shows that the SR process becomes a quality bottleneck, limiting the potential improvements in some scenarios. We considered whether this issue stems from neural networks themselves. Despite utilizing more powerful neural networks like RCAN [20] for testing, the enhancements are not significant, for about 26% of video chunks, the quality gain is lower than 3dB. Moreover, it introduces greater computational latency.

The commonly used low-pass filtering technique for downscaling [21], [22], as guided by the Nyquist–Shannon sampling theorem, inherently results in the loss of high-frequency details during the downscaling process [23], [24]. Given that SR tasks are fundamentally ill-posed and high-frequency details are more difficult to recover, relying solely on neural network optimization during the decoding phase is insufficient for achieving high-quality video output [25], [26].

To achieve this, we use a specialized neural network architecture to generate LR videos that diverge from traditional interpolation methods (§III-A). Furthermore, we have identified a significant opportunity to enhance the decoding process by leveraging inter-frame information, which has been underutilized in conventional SR techniques. We can obtain additional high-frequency information from the already decoded frames to assist in restoring the anchor frames (§III-C).

2) From Inter Frame's Perspective: Furthermore, our evaluations revealed that the decoding approach that reuses residuals from previously decoded frames incurs a degradation in quality. This decline comes from differences in how each method handles resolution adjustments during playback. Traditionally, to upscale LR videos for HR display, interpolation methods are used for each frame rather than for residuals. SR-integrated codec relies on interpolating residuals, despite utilizing information from already decoded frames, which also leads to an accumulation of errors brought by interpolation.

Fig. 4b illustrates the result of per-frame quality under the SR-integrated codec and traditional decoding methods following the SR of the first frame. Initially, the SR-integrated codec benefits from the HR first frame, yielding better quality in subsequent frames. However, as the video progresses, by 5_{th} frame, a decline in quality is observed with the SR-integrated codec compared to traditional decoding.

The error from reusing decoded frames is caused by two parts: the first part comes from the inaccuracies in SR on the anchor frames, and the second part comes from the accumulation of errors through continuous reuse.

The accumulation of errors arises from the imprecision of residual interpolation, and this decline is attributed to the cumulative effect of errors that build up over time. To address this issue, we revealed the relationship between inter-frame content and the decline in reuse quality in §Sec. III-B, and designed a new neural network training method that adjusts the overall content to reduce error accumulation without affecting the quality during the compression process.

Overall, existing systems have flaws in both Intra-frame and Inter-frame aspects. Given these challenges, we recognize that in video-on-demand streaming scenarios, we could control over the entire video streaming process, which raises the question: why not further optimize the cloud-side video encoding process? Existing solutions have only addressed mobile-side decoding processes, while lacking effective analysis on how to adapt cloud-side operations for neural-enhanced video-on-demand streaming. Therefore, we propose to collaboratively optimize the existing system on both the cloud side and the mobile side.

III. System Design

To overcome the limitations of previous works in Sec. II-B, we propose DoMo to enhance video quality through co-design of the cloud-mobile. Fig. 3 shows the architecture of DoMo. Our design includes four modules:

(i) Invertible Neural Network efficiently encodes high-frequency information to LR videos.

(ii) Reuse Aware Neural Network Training perceives the video quality loss caused by interpolation of frames during the upscaling process, thereby enhancing the video quality.

(iii) Video Frames Joint Selector selects reference frames, core frames, and non-core frames to decrease the overall error when reusing the decoded frames.

(iv) Ref-based neural decoder reconstructs anchor frames based on the existing SR-integrated decoder.

A. Efficient Invertible Neural Network Design

Our goal is to concurrently train the downscaling and upscaling processes in videos using neural networks, aiming to enhance the quality of reconstructed videos. Traditional neural-enhanced video streaming systems [13], [14], [27] focused primarily on developing neural networks for upscaling, neglecting the downscaling aspects.

Drawing inspiration from pioneering efforts in image enhancement [24], [28], we have incorporated the Haar wavelet transform along with an invertible neural network to form the backbone framework of our neural network processing pipeline (Fig. 5). To enhance efficiency, we have refined the design of the original reversible scaling network. we first apply downscaling modules before running any inverse block to enable the reversible convolutional layers to be applied on lower resolutions. We also decrease the depth and complexity of reversible convolutional layers.

Specifically, for inputs and outputs on a scale of s, we employ the Haar transform to convert the HR input (with dimensions $H \times W$) into low-frequency content l_0 (with dimensions $\frac{H}{s} \times \frac{W}{s} \times 3$) and high-frequency details h_0 (with dimensions $\frac{H}{s} \times \frac{W}{s} \times 3 \times (2^s - 1)$). Then these components are refined through invertible convolutional layers. Noticing that the matrix of high-frequency information is sparse, first we use an invertible layer to integrate high-frequency information and reduce the number of operating channels. We then use another invertible network layer to encode high-frequency information into low-frequency information. The detail of the forward process of invertible block is applied to the definition in [29]:

$$\begin{aligned} l_{i+1} &= l_i + \phi(h_i), \\ h_{i+1} &= h_i \odot \exp(\rho(l_{i+1}))) + \eta(l_{i+1}) \end{aligned} \tag{2}$$

where ϕ, ρ, η are functions defined by the neural network, and here is defined by the convolutional block, and $\odot$ is the element-wise product. The processed low-frequency frame content is quantized in `uint8` format to storage.

During the decoding phase, we employ a similar structure to compute the inverse process of the Equation 2 for upscaling and enhancing mobile-side video frames.

Our experiments demonstrate that this innovative network architecture significantly enhances the restoration of video quality, outperforming traditional models.

B. Reuse Aware Neural Network Training

In our training, we pursue three key objectives:

- To minimize the quality degradation of non-anchor frames which reuse the recovered HR frames during the video player process on the mobile side.
- To preserve the legibility of LR videos, ensuring that they remain as clear as when neural networks are not applied.
- To enhance the capability of the upscaling network, enabling it to more effectively recover details through LR frames transmitted to the mobile.

To fulfill these goals, our neural network training strategy is structured into three distinct parts.

Inter Frame loss: As discussed in Chapter 2, selectively executing neural networks has the cost that reusing frames restored by SR results in quality decline due to interpolation. Due to the residuals between frames (i.e., temporal differences), reusing SR results can lead to a reduction in quality. Moreover, since the video reconstruction process is a non-differentiable process involving the overall encoding process of the video, we cannot directly use gradient descent to precisely train the neural network to directly reduce the interpolation loss of non-anchor frame reconstruction.

We attempt to identify an optimization objective that enables us to enhance this process through neural network-based training. Inspired by the enhancement process of non-anchor frames, we discovered that the quality degradation caused by repeated anchor point usage exhibits a relatively strong linear correlation (Fig. 6a, $\rho = 0.732$) with residual values, while these residuals are highly correlated with pixel differences between adjacent frames (Fig. 6b, $\rho = 0.658$). This observation suggests that we can train neural networks to generate low-resolution videos with controlled inter-frame pixel differences to mitigate the quality degradation trend in non-anchor frames. Although this study primarily establishes a correlation rather than causality, it nonetheless suggests that pixel differences could be regarded a target for optimization during the network training process. This insight provides a promising direction for refining our approaches to improving video streaming quality.

Therefore, we use the pixel difference between LR frames as an additional loss:

$$L_{inter} = \frac{1}{M} \sum_{t=1}^{M-1} (\mathcal{L}(y_t, y_{t+1})), \tag{3}$$

where y_t is the LR frames generated by the downscaling neural network, M is the number of frames in a chunk and $\mathcal{L}$ is a difference metric like the mean square error.

HR reconstruction loss: Our aim is to maximize the quality of the reconstructed high-definition anchor frames. Therefore, we minimize the differences between the original

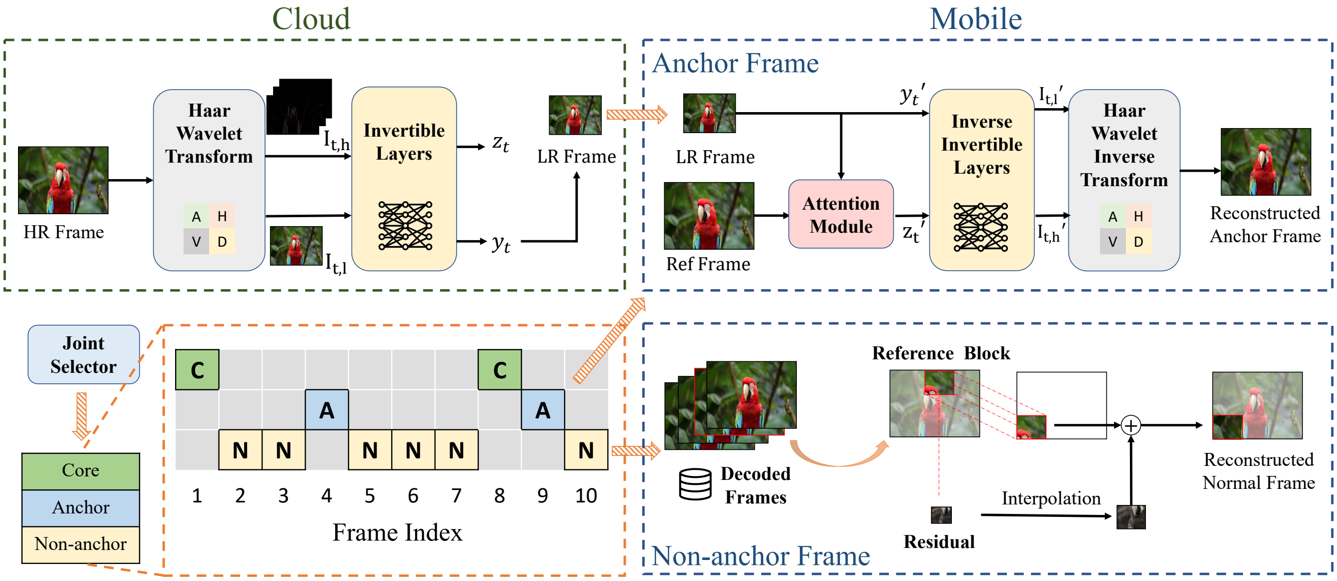
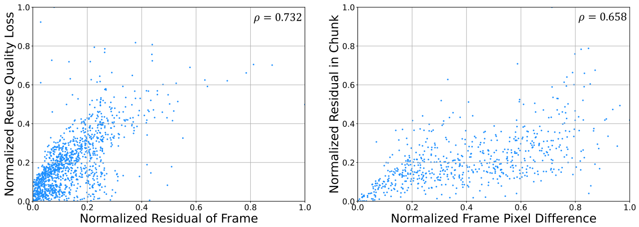


Fig. 5: Frame Process Pipeline



(a) (b)

Fig. 6: (a) Positive relation between the residual and the quality dec. (b) Positive relation between the residual and the average pixel difference in a video chunk

high-definition frames and the reconstructed high-definition frames. The HR reconstruction loss is defined as

$$L_{hr} = \frac{1}{M}\sum_{t=1}^{M}(\mathcal{L}(I_t', I_t)), \tag{4}$$

where I_t' is the reconstructed HR frame generated by the upscaling network (Fig. 5).

LR reference loss: The above two objectives ensure high-quality recovery effects under SR conditions. However, we aim for our system to maintain good video readability and quality even when devices lack SR capabilities. Therefore, we employ a LR reference loss based on interpolation, defined as follows:

$$L_{lr} = \frac{1}{M}\sum_{t=1}^{M}(\mathcal{L}(\uparrow y_t, I_t)), \tag{5}$$

where I_t is the original HR frame, $\uparrow$ is an interpolation method. In our experiment, $\uparrow$ uses the bilinear interpolation.

Overall, the loss function is combined with the three parts:

$$L = \lambda_1 L_{inter} + \lambda_2 L_{hr} + \lambda_3 L_{lr}. \tag{6}$$

C. Ref-based Neural Decoder

In our redesign of the encoding process, we introduced core frames specifically designed to retain HR details. Specifically (refer to Fig. 5), Core frames are transmitted in HR to preserve critical HR detail within the video. Anchor frames are transmitted in LR, with SR algorithms applied on the client side to enhance the video frame quality. Non-anchor frames are also transmitted in LR but are quickly reconstructed using non-neural network algorithms.

We developed the Ref-based SR Decoder to refine the workflow of the decoder and maximize the utilization of these high-definition details across more frames. We design a lightweight attention module to extract high-frequency details from the decoded core and anchor frames. In addition, we implemented a reference scheduling method that strategically selects previously decoded frames for reuse.

Cache Management: we extend the reference buffer within the VP9 codec to support the Ref-based Neural Decoder. In terms of cache management, frames are dynamically added to the cache as they are decoded. Non-anchor frames are subsequently removed from the cache once they are no longer needed as references for future frames. Core and anchor frames are retained in the cache until they are no longer needed for reference by any non-anchor frames and are not scheduled for reuse in enhancing anchor frames. Upon receiving a video chunk, the mobile begins the sequential decoding process and frames are classified based on the configuration file.

Core Frames are directly decoded into HR frames, ensuring premium visual quality.

Anchor Frames decoding process starts by reading the reference frame information from the configuration file and retrieving the decoded frame data from the high-level cache. Subsequently, a lightweight attention neural network is used to extract and align high-frequency information relevant to the

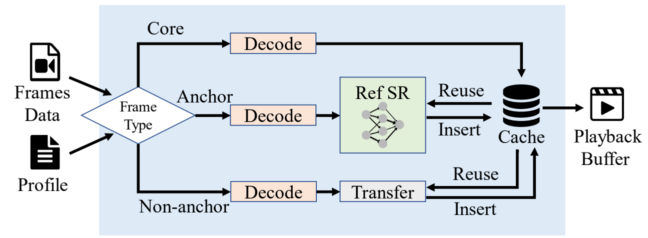


Fig. 7: Ref-based Neural Decoder System

current frame from the reference frames, followed by the use of an invertible neural network for image SR (Fig. 5).

Non-anchor frames are decoded by the SR-Integrated decoder. Firstly, we read the reference block, motion vector, and residual information about the block in the frame through the encoding information. Subsequently, utilizing motion vectors, the decoder aligns the target block from the reference block. Finally, the decoder employs lightweight bilinear interpolation to decode and upscale the residual, which is then accumulated onto the transferred blocks to output HR blocks.

D. Video Frames Joint Selector

During the offline optimization phase, our goal is to strategically select Core and Anchor frames within acceptable latency limits to enhance video decoding quality.

However, a significant challenge arises in assessing the potential of each frame to serve effectively as a Core frame. Since each frame could be a Core, Anchor or Non-anchor frame, the whole selection space is $O(3^{|Frames|})$, and it is computationally infeasible to evaluate every possible configuration through real SR tests.

According to the design of the DoMo system, the difference between the recovered video frames and the original video frames is determined by the discrepancy in their high-frequency information. Since adjacent frames possess similar high-frequency information, we can approximate the impact of the SR algorithm results by evaluating the gap in high-frequency information. Simultaneously, the extraction of high-frequency information can be synchronized with the downscaling process, eliminating the need for additional neural network operations and thereby reducing the overall computational overhead. We have

$$\begin{aligned} FQ(A \cup i, C)_k = \max_{j \in A \bigcup C, j<i} &(FQ(A,C)_k, \\ &(1 - \frac{\lambda \|z_i - z_j\|}{\|z_i\|}) FQ(i,)_k), \end{aligned} \tag{7}$$

where A is the set of anchor frames, C is the set of core frames, z_i is the high-frequency information generated by the downscaling neural network, and $FQ(\cdot,\cdot)_k$ is the quality of the frame k under the certain anchor and core frame selection.

Another issue that requires approximation is how to assess the impact of introducing core frames, which can cause other frames to lose quality at the same average video bitrate. It is impractical to re-encode the video with a new bitrate for every possible selection of core frames. Because the size change introduced by core frames could be calculated in advance, we can determine the value to which the bitrate of other frames will decrease and then estimate the overall change in video quality. The quality is as written:

$$FQ(A, C \cup i)_k = \frac{q(R')}{q(R)} \max(FQ(A,C)_k, FQ(,i)_k) \forall k \neq i, \tag{8}$$

where R and R' is the bitrate under the core selection of C and $C \cup i$, $q(R_n)$ is the utility of the chunk n. we utilized a linear bitrate utility function to quantify video utility, where $q(R_n) = R_n$ [27], [30].

Based on the above content, we can design a dynamic programming-based algorithm for selecting core frames and anchor frames, as well as choosing reference frames for SR of anchor frames. The selection result will be stored in the cache profile and transmitted to the mobile for decoding. The algorithm is as Algorithm 1.

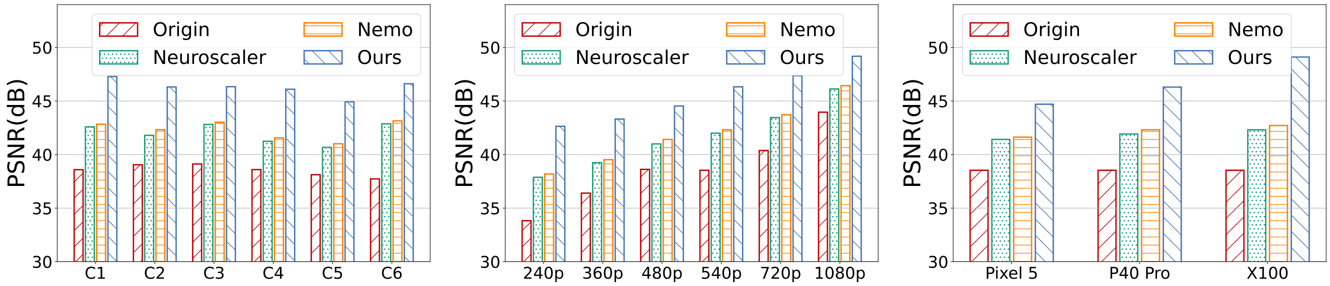
Algorithm 1 Joint Selector Algorithm

```
INPUT: Frames $F = \{F_i\}$, Max Anchor Num $M_a$, Max Core Num $M_c$, Frame Num $M$
OUTPUT: Anchor Set $A$, Core Set $C$, reference relation $\{r_{ij}\}$
for $i = 1$ to $n$ do
    $\{FQ(i,)\} \leftarrow$ RUN_SR_DECODER($F_i$)
    $\{FQ(,i)\} \leftarrow$ RUN_CORE_DECODER($F_i$)
end for
$VQ[0,1,0] = FQ(0,)$
for $i = 0$ to $M - 1$ do
    for $j = 1$ to $min(M_a, i)$ do
        for $k = 0$ to $min(M_b, i - j)$ do
            $VQ_{core} \leftarrow$ PREDICT_QUALITY_CORE($VQ[i-1, j, k-1], FQ$)
            $VQ_{anchor} \leftarrow$ PREDICT_QUALITY_SR($VQ[i-1, j-1, k], FQ$)
            $VQ_{other} \leftarrow VQ[i-1, j, k]$
            $VQ[i,j,k] \leftarrow \max\{VQ_{core}, VQ_{anchor}, VQ_{other}\}$
            $A, C, \{r_{ij}\} \leftarrow \arg\max\{VQ_{core}, VQ_{anchor}, VQ_{other}\}$
        end for
    end for
end for
```

IV. Implementation

Our system builds upon the open-source Nemo [13] framework, incorporating enhancements to the encoding and decoding functions within libvpx [18] for the VP9 codec [19]. For neural network operations, we utilize the TensorFlow Lite framework [31], chosen for its efficiency and compatibility with mobile platforms. The video data are stored in the YUV format. For neural network processing, we convert the video frames from the YUV420 format to the RGB888 format.

To better preserve the quality of core frames while improving video processing efficiency, we adopt an approach similar to BiSR [16], where high-resolution frames are separately stored as a video segment and transmitted alongside the



(a) Quality Comparison

(b) Various Resolutions Experiment

(c) Various Devices Experiment

Fig. 8: Main Result

TABLE I: Information of Mobile Devices

Devices	Processor	SR Latency
Google Pixel 5	Snapdragon 765	1157ms
Huawei P40 Pro	Kirin 990	487ms
Vivo X100	Dimensity 9300	114ms

low-resolution video. Specifically, this high-resolution video segment contains only core frames while maintaining the original bitrate of high-resolution video. After transmission to mobile devices, these high-resolution frames are prioritized for decoding to accommodate reference-based decoding methods.

V. Evaluation

A. Experiment Setting

Mobile Devices: To assess our system's performance, we conducted experiments on three smartphones with varying hardware configurations. These were the Google Pixel 5, the Huawei P40 Pro, and the Vivo X100. Table I highlights the specification details of these devices.

Video Dataset: For our experimental setup, we curated a collection of 18 videos from YouTube, representing a diverse range of content types that are popular among viewers [32], [33]: Product Reviews (C1), Video Tutorials (C2), Vlogs (C3), Gaming (C4), Music (C5), and Reviews (C6). Each video is in high-quality 4K (2160p) 30fps format using the VP9 codec, and we extracted five minutes from each for the experiment.

To assess the performance of our system under various conditions, we downscale each video to {240p, 360p, 480p, 540p, 720p, 1080p}, and set the bitrates respectively at {512, 1024, 1600, 2000, 2640, 4400} kbps. Unless specified otherwise, by default, we use a 540p video upscaled to 2160p for experiments, with the 4K video serving as the reference for video quality. In the experiments, the GOP (Group of Pictures) is set to 120, and the chunk length is 4 seconds.

Baseline: We compared our system with three methods:

- Origin: We do not perform any additional operations, simply decoding video frames normally and using bilinear interpolation to generate SR decoded frames.
- Nemo [13]: Due to space limitations, we conducted experiments using the high-quality neural network model described in Nemo.
- Neuroscaler [14]: Neuroscaler is a video streaming system designed for live streaming scenarios. In our experiments, we used Neuroscaler's anchor frame selection and decoding modules. We conducted tests using the same neural network as Nemo.

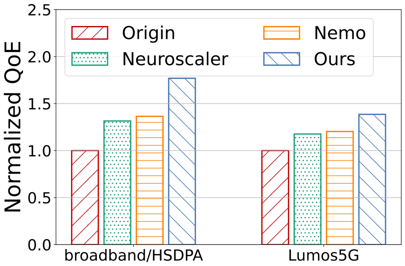
DNN Setting: To ensure a fair comparison environment, we adjusted the complexity of our neural network to guarantee a computational latency similar to that of the neural network used in NEMO [13] (within plus or minus 10% and measured by AI-benchmark [34]). For DNN training, we set $\lambda_1 = 1, \lambda_2 = 8, \lambda_3 = 1$. In our experiments, unlike NEMO [13] and BiSR [16], which dynamically adjust the size of the neural network, we fix the complexity of the neural network to facilitate a more equitable comparison.

B. Main Results

In our initial evaluations, we focused on the effects of quality enhancement across a range of videos, standardizing the conditions by setting the video stream to 540p at 2000 kbps. Fig. 8a shows the result of the quality of the video with a consistent decoding throughput of 30 fps, using various upscaling methods. Our approach markedly surpassed both Nemo and Neuroscaler in enhancing video quality in all tested categories. Compared to Nemo, there was an improvement in video quality of 3.32 - 4.54 dB in different video categories.

In our comprehensive tests across a range of transmission resolutions, we evaluated the performance of our upscaling method by restoring videos from lower resolutions (240p, 360p, and 480p) to 1080p and higher resolutions (540p, 720p, and 1080p) to 2160p (Fig. 8b). Our findings clearly demonstrate that our method consistently delivers significant enhancements in video quality across all tested resolutions (5.22-8.81 dB compared with the origin). In particular, the improvements became more pronounced as the scale between the original resolution and the target was increased.

For different devices, we used the same neural network and adjusted the number of anchor frames to ensure consistent throughput maintained at 30fps for testing. Our experiments demonstrate that regardless of the level of the device used, our method exhibits significant superiority. For devices with higher computing power, our algorithm performs better. Greater com-



(a) QoE Comparison

Fig. 9: QoE Evaluation

puting power means that more neural networks can be used, thus better unleashing the potential of our neural networks.

We conducted tests under real network conditions from broadband [35], HSDPA mobile network datasets [36] and 4G network trace in Lumos5G [37] dataset. As the design of our system is orthogonal to ABR algorithm optimization, we selected a fixed ABR method for this experiment to evaluate the performance of our system. We used Pensieve [9] as the ABR method. In the baseline scenario (Pensieve), we did not apply SR, transmitting the video directly. In other scenarios, we optimized the transmission of LR videos and applied SR for mobile-side enhancement. We conducted a comparative analysis using the average Quality of Experience (QoE) for each video category using the Pensieve method as a baseline.

To assess the impact of our system comprehensively, we adopted a widely recognized definition of Quality of Experience (QoE) [27], [38] as follows:

$$QoE = \frac{1}{N}(\alpha \sum_{n=1}^{N} q(R_n) - \beta \sum_{n=1}^{N} T_n \\ - \gamma \sum_{n=1}^{N-1} \|q(R_{n+1}) - q(R_n)\|), \tag{9}$$

where N is the number of video chunks, T_n is the rebuffer time, R_n is the bitrates, and the $q(R_n)$ is the utility of the chunk n. Additionally, we employed structure similarity index measure (SSIM) [39] to estimate the utility of videos that have undergone SR processing [27]. We set $\alpha = 1, \beta = 4.3, \gamma = 1$ for broadband and HSDPA mobile network datasets, and set $\alpha = 1, \beta = 20, \gamma = 1$ for Lumos5G dataset. The results, depicted in Fig. 9a, unequivocally demonstrate the superior performance of our system. In broadband and HSDPA datasets, our method achieved improvements of 76.9% and 29.6% respectively compared to the vanilla Pensieve and Nemo approaches. On Lumos5G, while the enhancement effects of video frames were relatively less pronounced due to higher network speeds supporting low-resolution videos at higher frame rates, our method still demonstrated significant improvements of 38.5% and 15.3%.

C. Ablation Study

We assessed the contributions of three pivotal modules: Neural Downscaling, Hierarchy Codec & Joint Selector, and Ref-based Decoder. This analysis was framed against the backdrop of the existing Nemo [13] system to gauge relative performance improvements. Our findings reveal each part of the design benefits the performance of the system.

We evaluate the neural network architectures within our video streaming system. All experiments use exactly the same settings except for the downsampling neural network. Fig. 10b indicates that our downscaling architecture based on invertible neural networks is beneficial for overall video enhancement, showing a 1.3 dB improvement in video quality compared to using a simple CNN-based downscaling neural network.

D. Inter Loss Analysis

In the process of our neural network training, we incorporated the inter-frame loss part. The design aims to make the video frames more coherent with each other, thereby reducing the quality decline when reusing decoded frames. Fig. 10c shows the impact of inter-frame loss. It reduces the video quality in per-frame SR, but improves the video quality in Selected SR. This underscores the importance its value, because Per-frame inference on mobile devices is inefficient.

In addition, as shown in Fig. 10d, the benefits of incorporating inter-frame loss are evident, particularly in maintaining quality over time. Without this feature, the initial high-quality anchor frame's impact quickly diminishes, eventually performing worse than a standard Nemo-processed frame. By integrating inter-frame loss, we substantially curb this quality decay, ensuring a more consistent and higher overall quality and significantly reducing fluctuations.

VI. Discussion and Futurework

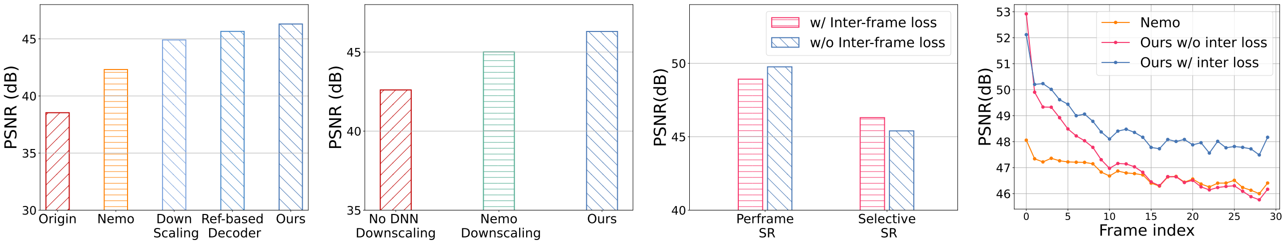
A. Codec Support

Our video streaming system is initially optimized for the VP9 [19] codec. However, we may encounter different encoding methods, such as H264 [40], AV1 [41] and other neural-based codec [42]–[45]. These codecs all use residuals and motion vectors to store compressed video, so our Ref-based Decoder can support them well. From an encoding perspective, our server's design does not depend on how each frame is specifically encoded, but rather optimizes the content of the downscaling process and the encoding format. Overall, we believe that our method can easily be migrated to other encoding methods and can be expected to achieve good results.

Some recent works propose codec design using deep learning [43]–[46]. This encoder is designed to retain superior video quality at the same bitrates. Further consideration could be given to the consideration of downscaling and compression simultaneously during the neural encoding process.

B. Combined with ABR Algorithm

Our existing system operates independently of the ABR module, suggesting that our approach can be integrated with any existing ABR [8]–[10], [47] method to achieve improvements in video quality. However, our system design leverages cloud-based video downsampling and re-encoding processes, while neural network-based video streaming exhibits different quality distributions compared to conventional approaches.



(a) Component-wise Result (b) Downscaling Method Result (c) Inter-frame Loss Comparison (d) Inter-frame Loss Comparison

Fig. 10: Ablation Study and Inter-frame Loss Study

Therefore, we believe that further integration of our solution with ABR algorithms, particularly considering the joint frame selector's bandwidth budget for different frame types, would be beneficial for additional system optimization.

VII. Related Work

A. *Video Streaming and Adaptive streaming*

Video streaming technology is a technique for continuous transmission and playback of video data over networks. As one of the core technologies in modern internet video transmission, it is extensively deployed in various video scenarios, including drone perception and video processing [48]–[50], vehicular perception and video analysis [51], and media video playback [13], [14], [16]. Through technologies such as video segmentation, continuous transmission, and adaptive bitrate streaming, video streaming technology can optimize bandwidth utilization and meet the requirements for real-time video interaction or analysis.

Adaptive bitrate streaming is one of the core technologies in video stream transmission and is designed to optimize user viewing experiences and adapt to fluctuating network conditions [6], [7], [52]. Each video is segmented into chunks and encoded at various bitrates and resolutions, creating multiple versions. Mobile devices, such as smartphones, are equipped with an adaptive bitrate (ABR) algorithm [8]–[10], [47] to dynamically adjust the quality of the streaming.

Our system's approach to optimizing video quality is orthogonal to that of adaptive streaming. Therefore, our method uses computational resources on the mobile side to further enhance service quality beyond adaptive streaming.

B. *Super Resolution (SR)*

SR technology often utilizes neural network techniques to enhance the resolution of images or videos [11], [28], [53]. It involves processing LR content and generating versions of higher resolution than the input images or videos. Due to the inherent requirement to process larger resolution feature layers, real-time applications on mobile devices encounter significant trade-offs between computational efficiency and enhancement quality [11], [12].

SR fundamentally addresses an ill-posed problem: Multiple HR results can be downscaled to the same LR image [53], [54]. Therefore, with the presence of ground truth for upscaling, some researchers have explored the strategy of executing the downscaling process with neural networks, which are then jointly trained with upscaling networks [24], [28]. However, these approaches have not yet been optimized for mobile devices, nor have they been re-designed for video streaming services.

C. *Neural-enhanced Video Delivery System*

The primary paradigm for neural-enhanced video delivery systems involves transmitting low-resolution content, followed by enhancement on the mobile side through super-resolution models. This approach utilizes mobile computing resources and opens a new dimension in scheduling space.

Researchers are exploring the use of neural networks to enhance video stream quality in various domains, including on-demand video streaming [13], [16], [27], live streaming [14], 360° panoramic videos [55], volumetric videos [15] and video conference [56]. In the domain of on-demand streaming, NAS [27] enhances quality frame-by-frame using a lightweight DNN. NEMO [13] suggests performing neural network enhancements only on a subset of frames. BISR [16] involves super-resolving only the first frame of the chunk while maintaining the other frames at a higher resolution. However, this method does not fully utilize mobile-side resources and is not compatible with existing CDN infrastructures.

VIII. Conclusion

In this paper, we present DoMo, the first work to rethink and unleash the potential of mobile-side video super-resolution from the cloud perspective. DoMo enhances video quality through a closed-loop neural-enhanced on-demand video streaming system. Extensive evaluations demonstrate that our system achieves improved video quality, QoE, and power consumption performance.

Acknowledgment

This work is supported in part by the National Key R&D Program of China under grant No. 2024YFC2607400, and the Natural Science Foundation of China under Grant No. 62202263, 62232004, 62302254, 62332016, 62472369, 62302259, and Tsinghua University-Fuzhou Joint Institute for Data Technology.

References

[1] "10 Video Marketing Trends To Watch In 2024 Based On Industry Experts," https://marketsplash.com/video-marketing-trends/.

[2] "4k technology market." [Online]. Available: https://www.gminsights.com/industry-analysis/4k-technology-market

[3] "Video Streaming Market Size, Global Trends Analysis, 2024-2032." [Online]. Available: https://www.polarismarketresearch.com/industry-analysis/video-streaming-market

[4] A. A. Laghari, S. Shahid, R. Yadav, S. Karim, A. Khan, H. Li, and Y. Shoulin, "The state of art and review on video streaming," *Journal of High Speed Networks*, vol. 29, pp. 211–236, 2023.

[5] A. Pimpinella, A. Marabita, and A. E. Redondi, "Crowdsourcing or network kpis? A twofold perspective for QoE prediction in cellular networks," in *Proc. of the IEEE WCNC*, 2021, pp. 1–6.

[6] "Dash industry forum," https://dashif.org/.

[7] "Apple http live streaming," https://developer.apple.com/streaming/.

[8] 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.

[9] H. Mao, R. Netravali, and M. Alizadeh, "Neural adaptive video streaming with pensieve," in *Proc. of the ACM SIGCOMM*, 2017, pp. 197–210.

[10] T. Huang, C. Zhou, R.-X. Zhang, C. Wu, X. Yao, and L. Sun, "Comyco: Quality-aware adaptive video streaming via imitation learning," in *Proc. of the ACM MM*, 2019, pp. 429–437.

[11] A. Ignatov, A. Romero, H. Kim, and R. Timofte, "Real-time video super-resolution on smartphones with deep learning, mobile AI 2021 challenge: Report," in *Proc. of the CVPR workshop*, 2021, pp. 2535–2544.

[12] S. Liu, C. Zheng, K. Lu, S. Gao, N. Wang, B. Wang, D. Zhang, X. Zhang, and T. Xu, "Evsrnet: Efficient video super-resolution with neural architecture search," in *Proc. of the CVPR workshop*, 2021, pp. 2480–2485.

[13] H. Yeo, C. J. Chong, Y. Jung, J. Ye, and D. Han, "Nemo: enabling neural-enhanced video streaming on commodity mobile devices," in *Proc. of the ACM MobiCom*, 2020, pp. 1–14.

[14] H. Yeo, H. Lim, J. Kim, Y. Jung, J. Ye, and D. Han, "Neuroscaler: Neural video enhancement at scale," in *Proc. of the ACM SIGCOMM 2022*, pp. 795–811.

[15] A. Zhang, C. Wang, B. Han, and F. Qian, "YuZu: Neural-Enhanced volumetric video streaming," in *Proc. of the 19th USENIX NSDI*, 2022, pp. 137–154.

[16] Q. Yu, Q. Li, R. He *et al.*, "Bisr: Bidirectionally optimized super-resolution for mobile video streaming," in *Proc. of the ACM WWW 2023*, pp. 3121–3131.

[17] A. Youssef, "Image downsampling and upsampling methods," *National Institute of Standards and Technology*, 1999.

[18] "Google's libvpx Official Github Repository," https://github.com/ webm-project/libvpx.

[19] "Vp9 video codec," https://www.webmproject.org/vp9/.

[20] Y. Zhang, K. Li *et al.*, "Image super-resolution using very deep residual channel attention networks," in *Proc. of the ECCV*, 2018, pp. 286–301.

[21] D. P. Mitchell and A. N. Netravali, "Reconstruction filters in computer-graphics," *Proc. of the ACM SIGGRAPH*, vol. 22, no. 4, pp. 221–228, 1988.

[22] "Ffmpeg filter," https://ffmpeg.org/ffmpeg-filters.html.

[23] "Nyquist–shannon sampling theorem wiki," https://en.wikipedia.org/wiki/Nyquist\%E2\%80\%93Shannon_sampling_theorem.

[24] M. Xiao, S. Zheng, C. Liu *et al.*, "Invertible image rescaling," in *Proc. of the ECCV*. Springer, 2020, pp. 126–144.

[25] D. Fuoli, L. Van Gool, and R. Timofte, "Fourier space losses for efficient perceptual image super-resolution," in *Proc. of the ICCV*, 2021, pp. 2360–2369.

[26] J. Liang, A. Lugmayr, K. Zhang *et al.*, "Hierarchical conditional flow: A unified framework for image super-resolution and image rescaling," in *Proc. of the ICCV*, 2021, pp. 4076–4085.

[27] H. Yeo, Y. Jung, J. Kim, J. Shin, and D. Han, "Neural adaptive content-aware internet video delivery," in *Proc. of the USENIX OSDI*, 2018, pp. 645–661.

[28] Y.-C. Huang, Y.-H. Chen, C.-Y. Lu, H.-P. Wang, W.-H. Peng, and C.-C. Huang, "Video rescaling networks with joint optimization strategies for downscaling and upscaling," in *Proc. of the CVPR*, 2021, pp. 3527–3536.

[29] L. Dinh, J. Sohl-Dickstein, and S. Bengio, "Density estimation using real NVP," in *Proc. of the ICLR*, 2017.

[30] X. Yin, A. Jindal, V. Sekar, and B. Sinopoli, "A control-theoretic approach for dynamic adaptive video streaming over HTTP," in *Proc. of the ACM SIGCOMM 2015*, pp. 325–338.

[31] "TensorFlow Lite." [Online]. Available: https://www.tensorflow.org/lite

[32] "15 popular types of youtube videos in 2023 [expert recommendations]," https://databox.com/popular-youtube-video-types.

[33] "24 Most Popular Types of YouTube Videos in 2024," https://visme.co/blog/types-of-youtube-videos/.

[34] "Ai-benchmark," https://ai-benchmark.com/.

[35] "Raw Data - Measuring Broadband America 2016," https://tinyurl.com/42ja3u5v.

[36] H. Riiser, P. Vigmostad, C. Griwodz, and P. Halvorsen, "Commute path bandwidth traces from 3g networks: Analysis and applications," in *Proc. of the 4th ACM MMSys*, 2013, pp. 114–118.

[37] A. Narayanan, E. Ramadan, R. Mehta *et al.*, "Lumos5G: Mapping and predicting commercial mmWave 5G throughput," in *Proc. of the ACM IMC 2020*, pp. 176–193.

[38] T. Huang, C. Zhou, X. Yao *et al.*, "Quality-aware neural adaptive video streaming with lifelong imitation learning," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 10, pp. 2324–2342, 2020.

[39] 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.

[40] "H.264 : Advanced video coding for generic audiovisual services," https://www.itu.int/rec/T-REC-H.264.

[41] "Av1 video codec," https://aomedia.org/av1/.

[42] Y. Cheng, Z. Zhang, H. Li, A. Arapin, Y. Zhang, Q. Zhang, Y. Liu, K. Du, X. Zhang, F. Y. Yan *et al.*, "Grace:loss-resilient real-time video through neural codecs," in *Proc. of the 21st USENIX NSDI*, 2024, pp. 509–531.

[43] O. Rippel, A. G. Anderson, K. Tatwawadi *et al.*, "Elf-vc: Efficient learned flexible-rate video coding," in *Proc. of the ICCV*, 2021, pp. 14 479–14 488.

[44] O. Rippel, S. Nair, C. Lew, S. Branson, A. G. Anderson, and L. Bourdev, "Learned video compression," in *Proc. of the ICCV*, 2019, pp. 3454–3463.

[45] M. Dasari, K. Kahatapitiya, S. R. Das, A. Balasubramanian, and D. Samaras, "Swift: Adaptive video streaming with layered neural codecs," in *Proc. of the 19th USENIX NSDI*, 2022, pp. 103–118.

[46] G. Lu, W. Ouyang, D. Xu *et al.*, "Dvc: An end-to-end deep video compression framework," in *Proc. of the CVPR*, 2019, pp. 11 006–11 015.

[47] F. Y. Yan, H. Ayers, C. Zhu, S. Fouladi, J. Hong, K. Zhang, P. Levis, and K. Winstein, "Learning in situ: a randomized experiment in video streaming," in *Proc. of the 17th USENIX NSDI*, 2020, pp. 495–511.

[48] H. Wang, J. Xu, C. Zhao *et al.*, "TransformLoc: Transforming MAVs into Mobile Localization Infrastructures in Heterogeneous Swarms," in *Proc. of the IEEE INFOCOM 2024*, pp. 1101–1110.

[49] H. Wang, X. Chen *et al.*, "H-SwarmLoc: efficient scheduling for localization of heterogeneous MAV swarm with deep reinforcement learning," in *Proc. of the 20th ACM SenSys*, 2022, pp. 1148–1154.

[50] X. Chen, Z. Xiao, Y. Cheng *et al.*, "SOScheduler: Toward Proactive and Adaptive Wildfire Suppression via Multi-UAV Collaborative Scheduling," *IEEE Internet of Things Journal*, 2024.

[51] Z. Jian, Z. Liu, H. Shao, X. Wang, X. Chen, and B. Liang, "Path generation for wheeled robots autonomous navigation on vegetated terrain," *IEEE Robotics and Automation Letters*, 2023.

[52] "Adobe http dynamic streaming," https://business.adobe.com/products/primetime/adobe-media-server/hds-dynamic-streaming.html.

[53] R. Lee *et al.*, "Deep neural network–based enhancement for image and video streaming systems: A survey and future directions," *ACM Computing Surveys*, vol. 54, no. 8, pp. 1–30, 2021.

[54] C. Dong, C. C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 2, pp. 295–307, 2015.

[55] M. Dasari, A. Bhattacharya, S. Vargas, P. Sahu, A. Balasubramanian, and S. R. Das, "Streaming 360-degree videos using super-resolution," in *Proc. of the IEEE INFOCOM*. IEEE, 2020, pp. 1977–1986.

[56] V. Sivaraman, P. Karimi, V. Venkatapathy, M. Khani, S. Fouladi, M. Alizadeh, F. Durand, and V. Sze, "Gemino: Practical and robust neural compression for video conferencing," in *Proc. of the 21st USENIX NSDI*, 2024, pp. 569–590.