

Application of machine learning for Managing High-Definition Video Streaming

Kumar Avinash*

**Software Development Engineer, Google ,Seattle, USA*

Abstract

The paper explores the idea of using machine learning for high-definition video streaming. Formats like 4K, 8K, and VR make this task harder because they need more stable delivery. The purpose is to review the work from 2021 to 2024 and apply different approaches — forecasting models, reinforcement learning, and some unsupervised methods — to see how they affect the stability and quality of adaptive bitrate streaming. The review covers a variety of research: forecasting with BiLSTM–CNN and GRU models, reinforcement learning systems like DQNReg, DeepVR, or GreenABR, and clustering techniques for monitoring QoE. The main results indicate that machine learning solutions outperform older ABR rules: fewer stalls, smoother quality, higher QoE, and in some cases lower power use on mobile devices. At the same time, the models are often heavy to run and may not generalize well outside the training data. The article is meant for researchers and engineers working on video delivery and network optimization, and points to where ML-based streaming could be applied in practice.

Keywords: machine learning; adaptive bitrate streaming; quality of experience; reinforcement learning; bandwidth prediction; unsupervised learning; video streaming; VR/360° video; energy-aware streaming; network optimization.

1. Introduction

Online video has grown so quickly that it now dominates internet traffic. The web has effectively turned into a bandwidth-hungry space. High-definition, 4K, and even immersive formats like 360° and VR make the problem worse, since they demand far more stable delivery. Streaming services therefore run into constant challenges: networks fluctuate, devices differ, and users still expect smooth, high-quality playback without pauses. Traditional adaptive bitrate (ABR) algorithms try to handle this by switching video quality according to buffer size or throughput estimates. That approach works to a degree but is easy to break.

Received: 9/19/2025

Accepted: 11/19/2025

Published: 11/29/2025

** Corresponding author.*

A sudden drop in bandwidth can trigger overreactions — long stalls, jarring quality changes, or wasted capacity when the network could have carried more. Machine learning has been proposed as a way out of this cycle. Predictive models, reinforcement learning, and even unsupervised methods make it possible to anticipate network changes instead of just reacting to them. Some of these methods aim at more than one target at once — they don't just cut stalls, but also try to hold resolution steady or, on phones, save a bit of battery. Not every improvement comes from ML, though. Systems-level fixes can help too. For example, Prime Video's Startover Playback, on which the author worked directly, tackled the problem of keeping live streams smooth on older or limited devices. Studies over the last few years show measurable gains in both objective terms (shorter stall times, fewer abrupt switches) and in subjective Quality of Experience (QoE). ML opens up things the old ABR algorithms were not able to do. One clear case is virtual reality (VR). In 360° video there is no necessity to send the entire scene in high resolution. Instead, the system can try to guess where the viewer will turn their head and keep that part sharp while lowering quality in the rest. That way, a lot of bandwidth is saved and most people will not notice the difference.

In general, machine learning lets streaming players do things the old ABR rules could not handle. It makes it possible to forecast bandwidth, adjust bitrate before problems occur, and tune playback to the special demands of immersive formats. A practical example is viewport prediction in 360° video, where only the visible part of the scene is sent in high quality, saving bandwidth without lowering user satisfaction. These kinds of methods show clear improvements in both playback stability and perceived quality. The studies from 2021 to 2024 examined in this paper illustrate how such approaches are being tested and refined in practice.

2. Methods and Materials

One study by Andruloniw and his colleagues used clustering for IPTV. No labels, just raw playback data, and it could still detect when quality was going down [1]. Dinaki and his colleagues went another way. They trained a BiLSTM-CNN on multivariate time-series, mixing temporal with spatial data. That let them predict QoE a step ahead and react before the stream froze [2]. Ghosh and Singhal tried to push accuracy further with MO-QoE, which blends network measurements and perceptual features [3]. Hou and his colleagues built NOVA for 360° streaming, designed to predict viewports and allocate bandwidth at the edge [4]. Kougiumtzidis and his colleagues didn't build a system but reviewed the field. Their conclusion: no single QoE metric is enough — the models need to combine several signals. Other engineering approaches, for instance the Startover Playback pipeline at Prime Video, have contributed to improved streaming resilience without ML by optimizing caching and time-indexed stream delivery

Other studies zoomed in on narrower bottlenecks. Lebreton and Yamagishi, for example, proposed a bitrate ladder adjustment based on the “quitting ratio,” a metric tied directly to user drop-off. Souane and his colleagues worked with deep reinforcement learning, building an LSTM-driven agent that adapted bitrates on the fly in wireless environments. Turkkan and his colleagues went in a different direction with GreenABR, which folded energy use into the reward function. Their tests showed that power consumption could be cut in half without lowering QoE. Woo and his colleagues combined a conventional buffer-based controller with GRU-based bandwidth prediction and reported about an 11% boost in QoE under unstable conditions. For VR, Xiao and his colleagues introduced

DeepVR, a reinforcement framework that predicts the user's viewport and adjusts bitrate accordingly, so only the visible part of the 360° scene is streamed in full quality [10]. Time-series forecasting has also become central. Dinaki and his colleagues showed that a BiLSTM–CNN model could anticipate QoE one step ahead by combining spatial and temporal features [2]. Woo and his colleagues applied a GRU predictor for bandwidth, which allowed bitrate changes before a drop was noticeable, again yielding an 11% gain compared with a plain buffer-based baseline [9]. These models rely on real network traces — throughput logs, stall events, bitrate histories — so they can act proactively rather than reactively.

Reinforcement learning has had a similar impact on adaptive bitrate (ABR) itself. Souane and his colleagues modified the state space with a “quality distance” feature, which let their LSTM-based agent make finer segment-by-segment decisions [7]. RL approaches handle adaptive bitrate (ABR) differently from heuristics. Instead of following fixed rules, they model it as a Markov Decision Process. The player makes a choice — say, a certain bitrate — then sees what happens to the buffer and the network, and adjusts the next step depending on the QoE reward. After enough rounds, the agent ends up with a policy that balances three things: keeping the resolution high, avoiding stalls, and keeping playback smooth. In trials with wireless traces, these RL agents consistently beat older buffer-based or rate-based schemes. The video looked steadier and interruptions dropped. Earlier literature—particularly the surveys by Kougioumtzidis and his colleagues and early ABR frameworks such as Pensieve—identified the need for composite QoE metrics and adaptive learning policies [5]. The present review extends that discussion through 2024 by showing that subsequent models operationalized these recommendations: MO-QoE fuses perceptual and network features, while GreenABR incorporates energy cost directly into the RL reward function [3, 8]. Together, these innovations reinforce and broaden the multidimensional idea of Quality of Experience described in earlier studies. In those earlier papers, machine learning was mostly a side feature—an add-on that helped heuristic controllers make slightly better guesses. More recent work turns that logic upside down: learning systems now form the core of streaming optimization itself. Taken as a whole, the shift traces a clear line from handcrafted bitrate rules to self-learning, multi-objective agents and links the theory work of 2020–2021 to the edge-computing frameworks that appeared around 2023–2024. By explicitly synthesizing these developments, the current paper provides a more integrated perspective on how machine learning transforms not only bitrate adaptation but the entire end-to-end design philosophy of high-definition video delivery. Nevertheless, much of the research cited here still comes from lab simulations or from a few narrow network-trace datasets such as LIVE-NFLX or the Norway traces. Those setups streamline the experimentation but also devalue the complexity of real networks, where congestion, codecs, and user interactions change from moment to moment. It is not unusual for a model that performs beautifully in those controlled runs to stumble once it meets live traffic. A second limitation is cost—deep models with millions of parameters can be awkward to deploy on lightweight or edge hardware that has to react within tight latency windows. To move forward, researchers could focus on testing across several datasets, trimming or distilling large networks so they run faster, and using live telemetry for continuous updates as conditions change.

3. Results and Discussion

Bitrate control has shifted a lot in recent years. Early work like Pensieve trained agents directly on QoE, and it worked, but stability was shaky. Later groups tried to fix that. Souane and his colleagues, for example, added

regularization to a DQN [7]. The change sounds small, but it sped up training and lifted QoE scores compared with older RL setups. Other researchers pushed toward actor-critic models. Soft actor-critic (SAC) allowed continuous bitrate choices rather than fixed steps, which made quality changes feel smoother on unstable links. A3C agents took a different route, adapting playback not just to the network but to user preference as well.

One approach is clustering streaming session data to detect QoE degradation patterns without explicit labels. Andruloniw and his colleagues developed an unsupervised two-phase clustering method to continuously assess QoE in an IPTV streaming system [1]. Their system analyzes metrics like video stall durations, quality switches, and time since last downswitch to group playback periods into “satisfaction” levels. This data-driven QoE assessment runs in real time on set-top boxes, helping network operators identify when a viewer’s experience deteriorates so they can adjust delivery (for instance, by reallocating bandwidth or switching CDN servers). Such unsupervised QoE monitoring is valuable for managing high-definition streams, as it can flag subtle quality issues (e.g. frequent small stalls or gradual quality drops) that might be overlooked by coarse metrics. It effectively creates an ongoing QoE “feedback loop” for the streaming service, which complements the proactive adjustments made by predictive and RL algorithms.

360-degree VR video streaming (an extreme case of high-definition streaming) presents additional challenges due to its enormous bandwidth demands. Machine learning has addressed the very special issues of panoramic and 360° video, wherein the bandwidth for transferring the full scene in high resolution becomes prohibitive. One common method is sending in high quality only that area which a viewer (the viewport) is looking at while reducing quality in areas not seen by viewers. This viewing direction prediction a few seconds in advance has become an important task. It was found by Ghosh & Singhal that the LSTM sequence model could predict a user’s viewport accurately enough to allocate higher resolution in advance to the relevant tiles [3]. Other systems use both spatial and temporal features; e.g., as in MOSAIC framework where a CNN is combined with an LSTM trained on prior head rotations, video motion fields, and saliency maps to anticipate where attention will shift and thus allocate bitrate.

Immersive streaming has also been tackled with learning approaches. One example is RAPT360, a DRL method that handles both viewport prediction and bitrate selection at the same time. It works by splitting the video into tiles and then using a buffer-aware scheme to guess which region the viewer is likely to look at next. The benefit is obvious. Less bandwidth is wasted on parts of the scene that nobody sees, while the visible viewport stays sharp. In practice this keeps the immersive experience stable without putting excessive strain on network resources.

An interesting development is energy-aware streaming for mobile devices, where ML helps minimize power consumption. A work by Turkkan and his colleagues introduced GreenABR, a deep RL-based ABR scheme that explicitly includes energy use in its reward function [8]. GreenABR’s neural agent was trained with perceptual quality (VMAF) and measured battery drain data, learning to make bitrate decisions that save energy while still keeping video quality high. In testing, GreenABR cut more than 50% of the energy used for streaming and about 60% of data usage, while also making QoE better by about 22% because there were fewer times when it needed to stop and buffer. This shows that reinforcement learning can optimize more than just perceptual quality. Turkkan and his colleagues, for example, balanced QoE against battery life, and their agent was able to satisfy both goals

at the same time. Later work pushed the idea further, adding other constraints — such as network traffic cost or fairness across multiple users on the same link.

In essence, these systems no longer treat QoE as the only target. As for the data side, most of these ML models are trained and tested on a mix of sources. Researchers often use real network traces to capture throughput variation. They also rely on standard test videos — Big Buck Bunny is a common one — since it can be streamed at different quality levels to check how the algorithm behaves. Finally, QoE datasets with subjective ratings are added in, so the models can be judged against how people actually perceive playback [7].

To illustrate the variability of the cellular traces used for evaluation, Figure 1 summarizes downlink throughput aggregated over 3-day windows.

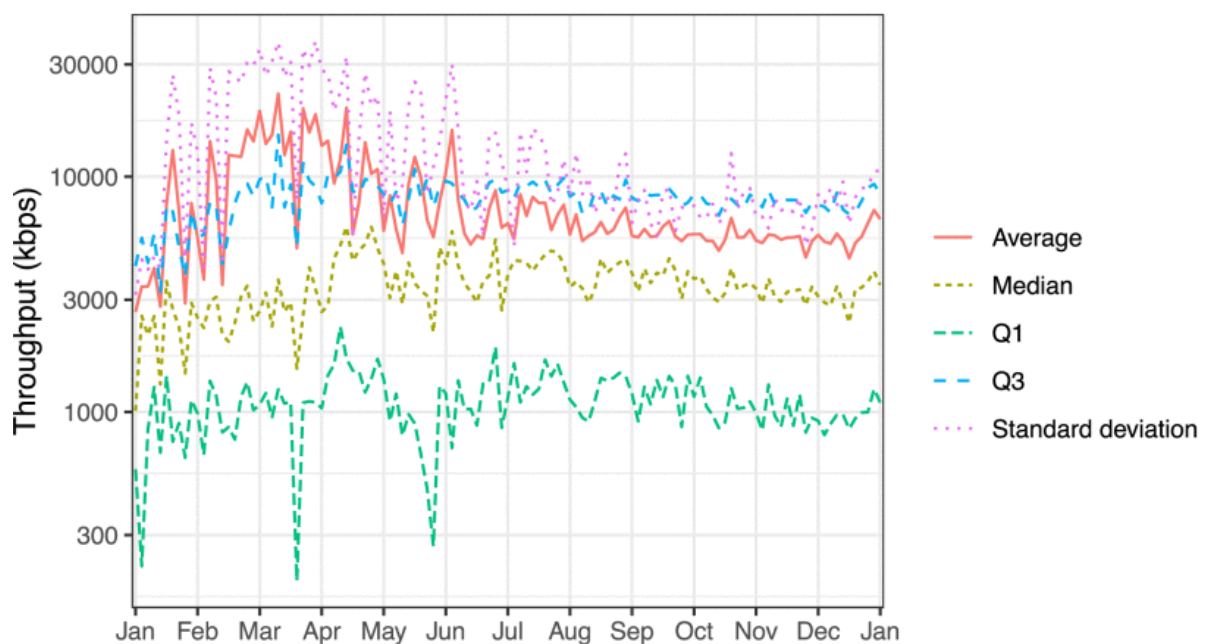


Figure 1: Distribution of throughput values across sessions conducted by Lebreton and Yamagishi [6]

Figure 1 shows wide dispersion from January to June ($Q1 \approx 300\text{--}600$ kbps vs. $Q3 \approx 6\text{--}8$ Mbps, mean $\approx 5\text{--}8$ Mbps), followed by a more stable second half of the year. The mean consistently exceeds the median, indicating right-skew with occasional high-throughput spikes. Because of the seasonally shifting and heavy-tailed distribution of throughput values, a bitrate controller tuned to a single “average” rate wastes capacity during peaks and also will stall during troughs. As noted by Lebreton and Yamagishi, such dynamics necessitate short-horizon prediction of bandwidth, and QoE that helps in predicting sudden drop as well as reinforcement learning policies that can balance bitrate selection, rebuffering penalty as well as switch cost under uncertainty [6].

Beyond the ML-based projects, the author also worked on Prime Video’s Startover Playback feature. This capability lets viewers on older or low-spec devices restart a live event from the beginning. The author designed and implemented key parts of the underlying stream pipeline, which combined smart caching with time-indexed references so that the feature could run smoothly inside the existing player. In practice, this reduced drop-offs,

made seeking more reliable, and drove much higher engagement during major broadcasts like Thursday Night Football. The project itself didn't involve machine learning, yet it turned out to be a good reminder that careful pipeline work can directly shape how people experience playback at scale.

Most supervised schemes for adaptive video still lean on subjective Quality-of-Experience scores—mainly mean-opinion ratings collected from viewer panels [2]. To build training and test sets that feel closer to what happens in the wild, researchers often fall back on open testbeds or shared trace libraries instead of fully synthetic samples. A common setup is dash.js with a few custom tweaks, or a Python simulator that replays hundreds of sessions pulled from real mobile network traces. Benchmarks matter too. The LIVE-NFLX dataset has become a default option that users reach for when they want to check if an ML-based adaptation really does smoother playback than the old ABR approaches. Figure 2 visualizes one playback's normalized feature trajectories over seven Δt steps, illustrating how short-term buffering dynamics track QoE.

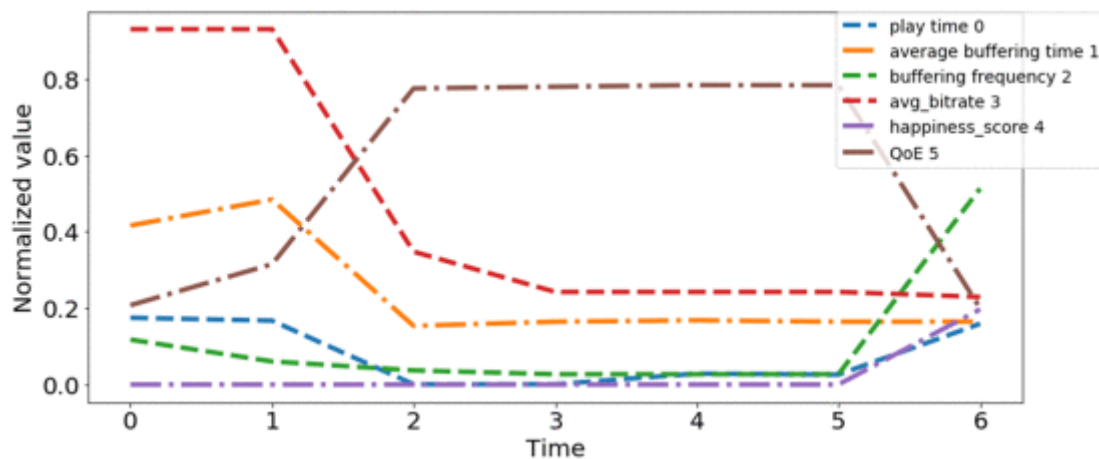


Figure 2: QoE Dynamics in a Single Playback by Dinaki and his colleagues [2]

Figure 2 shows one representative video session divided into seven Δt windows. Each curve is normalized to [0, 1] so that the feature trends are directly comparable. During stalls, the play-time curve (blue) dips, and the average buffering length (orange) collapses around $t = 2$, which temporarily lifts the overall QoE (brown). Later, at $t = 6$, buffering frequency (green) spikes and the QoE drops again even though the bitrate stays flat. The average bitrate (red) steps down early—typical adaptive-bitrate behavior under constrained bandwidth—and then stabilizes. Finally, the happiness score (purple) remains at zero until playback stops, since NPAW logs it only at the end. The analysis showed a clear trend. Viewers generally tolerate lower bitrates without much complaint, but they react strongly to stalls. Even a short pause can drag QoE down, while cutting stall time improves it right away. Similar gains were also achieved in practice through Startover Playback, which reduced user drop-offs by enabling late joiners to restart streams. Dinaki and his colleagues built on this observation when they classified sessions as “no issue,” “ISP issue,” or “client issue.” Their BiLSTM–CNN model could then use those categories to tell different types of degradation apart [2].

Machine learning has consistently been found through experiments to improve QoE across heuristic ABR controllers. To quote the work of Woo and his colleagues, their GRU-based bandwidth predictor increased average

QoE by approximately 11% in wireless tests with a reduced number of abrupt quality switches [9]. By predicting bandwidth drops and adjusting bitrate preemptively, sudden rebuffering events were avoided, too. Dinaki and his colleagues found that forecasting QoE one step ahead enabled prefetching at lower quality which kept the playback smooth and outperformed baseline prediction models [2].

Figure 3 visualizes training-time gains for the DRL ABR on the Norway trace set (70% sequences, real time), with convergence near hour 40 and saturation by hour 60.

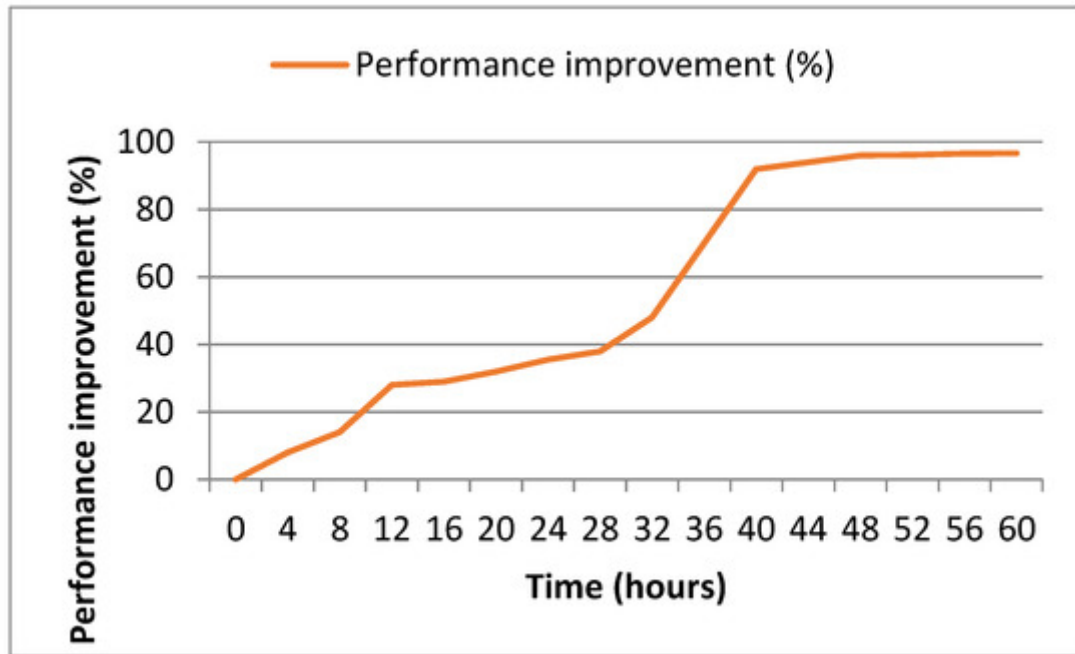


Figure 3: Performance improvement during training for the DRL-based ABR model conducted by Souane and his colleagues [7]

At the beginning the gains are very small. In the first eight hours the increase stays under 10%. Between the 8- and 12-hour mark the curve climbs quickly, reaching roughly 25%. From there it goes up more gradually, hitting around 40% by hour 32. There is a small bump between hours 32 and 36, and then the line jumps to about 90% near hour 40. Past that, the values flatten out and stay in the 90–95% range. Extending the training to 60 hours brings no real benefit. The test used 70% of the Norway trace set in real time, and convergence around hour 40 proved to be a practical cutoff: extra training only wasted computation without improving QoE.

Panoramic and 360-degree video streaming raises a different challenge — the entire scene cannot be sent at full resolution without overwhelming the network. In panoramic and 360° video, sending the whole scene in full resolution is not practical. Researchers such as Ghosh and Singhal showed that an LSTM model could guess the viewer's next field of view with around 80% accuracy. Using that prediction, the system streams the tiles in the likely viewing area at high quality and lowers the rest. This cuts down bandwidth use quite a bit, but the image still looks sharp in the part the viewer is watching. Other tile-based studies have confirmed the same pattern: if the non-visible regions are sent at lower quality, users generally do not notice. Reports show that viewport-aware

methods raise QoE in VR, since the viewed region stays crisp and the total bandwidth drops significantly. Reinforcement learning adds another layer. DeepVR, for example, learns how to balance present quality with the likelihood of a head turn. The result is smoother VR playback — sharp video where the viewer is looking, and, if they move, the next view is already buffered in high quality. Earlier ABR heuristics could not handle this. They ignored viewport shifts and treated QoE as a single objective, which left VR and ultra-HD streams full of artifacts. Machine learning methods fix that gap by predicting attention and reallocating bitrate. Still, there are drawbacks. Hou and his colleagues noted that their NOVA setup had to run inference in real time for every video chunk and used a model with millions of parameters, which is heavy. Reinforcement learning runs into the same issue: the system has only a few milliseconds to decide, which is taxing on normal hardware. To work around that, some teams shrink the models, while others shift the heavy lifting to edge servers.

Data, however, is another challenge. Supervised methods need huge labeled sets that tie network states to QoE results. RL approaches face a different issue, since they have to be trained on realistic simulations. Dinaki and his colleagues, for instance, built a BiLSTM–CNN on thousands of traces with QoE scores, but the accuracy still dropped once the model was tested on situations it had not seen before [2]. To counter this, newer work has tried domain randomization and online learning, letting models adapt when they encounter unfamiliar conditions. In practical terms, streaming providers may also return telemetry—chosen bitrate, stall events, and any available user feedback—to deployed models so as to make them self-improving systems eventually better and better over time. Kougiumtzidis and his colleagues found in their survey that a combination of even more indicators of QoE — among them startup delay, frequency of stalls, and consistency of resolution — gives the most accurate predictor of user satisfaction [5]. Ghosh and Singhal’s MO-QoE model combined perceptual features with network QoS data. By doing that, it lined up better with human MOS scores than models that relied on single inputs.

Looking across recent studies, a clear pattern emerges: machine-learning-based adaptive bitrate (ABR) controllers are moving the field from reactive toward predictive control of video streams. Researchers have tried different setups—BiLSTM-CNN predictors, GRU-style bandwidth estimators, and several reinforcement-learning agents—and most of them end up pointing in the same direction. Playback becomes smoother, and users report higher Quality of Experience even when the network keeps shifting [2; 9; 7; 8]. The observed gains probably arise from how these models follow temporal patterns and anticipate changes in available bandwidth while still trying to satisfy several, sometimes conflicting, quality goals. Forecasting approaches usually react in bursts to short-term drops or spikes, whereas reinforcement-learning ones evolve more slowly, adjusting their internal priorities as playback continues.

Each type captures a different rhythm of network behavior, and together they form the backbone of today’s adaptive-bitrate control logic. In practice, teams often combine the two: a GRU predictor takes care of bandwidth forecasts, and a DQN or actor–critic agent adjusts playback decisions on the fly. A similar pattern shows up in immersive and 360-degree streaming, where gaze-tracking modules guess what the viewer will look at next and assign bitrate accordingly, keeping the picture sharp while cutting data use by more than half [3; 4; 10]. All of this suggests that machine-learning controllers can work across many formats—linear, mobile, or VR—so long as the design accounts for temporal correlations and balances several objectives at once. There are still trade-offs that no tool has fully solved. Big architectures such as NOVA or DeepVR can reach the best QoE numbers, but they are

taxing on processing power and add small but noticeable delays. They also drain more energy than most setups can spare. Smaller recurrent models behave differently—they do not score as high, yet they are light enough to run on edge hardware without malfunctioning. In the end, practical success does not stem from prioritizing accuracy - it instead favors speed and power efficiency.

4. Conclusion

Recent research has already proven that machine learning will handily defeat legacy adaptive bitrate algorithms at the helm, when it comes to managing high-definition video delivery. Such forecasting models as BiLSTM–CNN of Dinaki and his colleagues improve QoE prediction which enables preemptive action against stalls before the actual stall takes place [2] while reinforcement learning controllers like DQNReg (Souane and his colleagues, 2023) offer even smoother playbacks and faster convergence compared to heuristic methods [7]. Special-use applications demonstrate this same trend: Hou and his colleagues use viewport prediction to make 360° streaming viable at scale [4], Turkkan and his colleagues reduce mobile power consumption by more than 50% with an energy-aware ABR agent [8]. These examples show that not only is ML-based streaming far more adaptive, but the flexibility goes much deeper in optimizing under disparate constraints. At the same time, serious limitations persist. As noted by Hou and his colleagues, the NOVA framework requires millions of parameters to execute in real-time, raising—indeed exploding—questions about computational feasibility on any resource-limited device [4]. Generalization is still a weak point. Dinaki and his colleagues [2] already showed that their model lost accuracy whenever it faced network conditions not covered in the training set. Subjective QoE scores add another layer of noise, since people do not always rate playback the same way. A further problem is fairness — especially when several users share the same link. That remains unresolved. For deployment, lighter models and edge-assisted inference look like the most realistic options. Another path is online learning from live telemetry, which would let the system adjust when it runs into situations that were never in the training data. Some groups also argue for “holistic” predictors that mix network signals with content features and even user behavior. And beyond the technical fixes it is not evident how to make these systems fair, and how to keep them sustainable over time. What is clear is that, with careful training and deployment, ML-based controllers deliver a better experience. They cut interruptions, smooth out playback, and use bandwidth more efficiently than any rule-based scheme so far. The next step is making these methods work with heavier formats like 4K, 8K, and even VR. If that doesn’t happen, systems risk becoming unstable or unfair in practice. On the other hand, not all progress comes from ML. An appropriate example is Prime Video’s Startover Playback—a feature built through engineering rather than modeling—that still led to raising engagement, smoothing playback, and keeping the service more accessible at scale. Overall, these complementary approaches indicate that advances in video streaming depend on the interplay between algorithmic innovation and solid system engineering.

References

- [1] P. Andruloniw, K. Kowalik, and P. Zwierzykowski. “Unsupervised learning data-driven continuous QoE assessment in adaptive streaming-based television system.” *Applied Sciences*, vol. 12(16), pp. 8288, Aug. 2022.

- [2] H. E. Dinaki, S. Shirmohammadi, E. Janulewicz, and D. Côté. "Forecasting video QoE with deep learning from multivariate time-series." *IEEE Open Journal of Signal Processing*, vol. 2, pp. 512–521, 2021.
- [3] M. Ghosh and C. Singhal. "MO-QoE: Video QoE using multi-feature fusion based optimized learning models." *Signal Processing: Image Communication*, vol. 107, pp. 116766, 2022.
- [4] B. Hou, S. Yang, F. Li, L. Zhu, X. Chen, Y. Wang, and X. Fu. "NOVA: Neural-Optimized Viewport Adaptive 360-Degree Video Streaming at the Edge." *IEEE Transactions on Services Computing*, vol. 17(6), pp. 4027–4040, 2024.
- [5] G. Kougioumtzidis, V. Poulkov, Z. Zaharis, and P. Lazaridis. "A survey on multimedia services QoE assessment and machine learning-based prediction." *IEEE Access*, vol. 10, pp. 19507–19538, 2022.
- [6] P. Lebreton and K. Yamagishi. "Quitting ratio-based bitrate ladder selection mechanism for adaptive bitrate video streaming." *IEEE Transactions on Multimedia*, vol. 25, pp. 8418–8431, 2023.
- [7] N. Souane, M. Bourenane, and Y. Douga. "Deep reinforcement learning-based approach for video streaming: Dynamic adaptive video streaming over HTTP." *Applied Sciences*, vol. 13(21), pp. 11697, 2023.
- [8] B. O. Turkkan, T. Dai, A. Raman, T. Kosar, C. Chen, M. F. Bulut, and D. Sow. "GreenABR: Energy-aware adaptive bitrate streaming with deep reinforcement learning." in *Proc. 13th ACM Multimedia Systems Conference (MMSys 2022)*, 2022, pp. 150–163.
- [9] J. Woo, S. Hong, D. Kang, and D. An. "Improving the quality of experience of video streaming through a buffer-based adaptive bitrate algorithm and gated recurrent unit-based network bandwidth prediction." *Applied Sciences*, vol. 14(22), pp. 10490, Nov. 2024.
- [10] G. Xiao, M. Wu, Q. Shi, Z. Zhou, and X. Chen. "DeepVR: Deep reinforcement learning for predictive panoramic video streaming." *IEEE Transactions on Cognitive Communications and Networking*, vol. 5(4), pp. 1167–1177, Dec. 2021.