

## Research Article

# A Novel Model for Large-Scale Online College Learning in Postpandemic Era: AI-Driven Approach

Cong Wang 

Jinzhou Medical University, Jinzhou 121000, China

Correspondence should be addressed to Cong Wang; wangcong410@jzmu.edu.cn

Received 25 October 2021; Revised 8 November 2021; Accepted 1 December 2021; Published 14 December 2021

Academic Editor: Jianhui Lv

Copyright © 2021 Cong Wang. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

COVID-19 is a pandemic with a wide reach and explosive magnitude, and the world has been bracing itself for impact. Many have lost their jobs and savings, and many are homeless. For better or worse, COVID-19 has permanently changed our lives. For college students, the pandemic means giving up most of the on-campus experience in the postpandemic era and performing online learning instead. Virtual lessons may become a permanent part of college education. Large-scale online learning typically utilizes interactive live video streaming. In this study, we analyzed a codec and video streaming transmission protocol using artificial intelligence. First, we studied an intraframe prediction optimization algorithm for the H.266 codec based on long short-term memory networks. In terms of video streaming transmission protocols, real-time communication optimization based on Quick UDP Internet connections and Luby Transform codes is proposed to improve the quality of interactive live video streaming. Experimental results demonstrate that the proposed strategy outperforms three benchmarks in terms of video streaming quality, video streaming latency, and average throughput.

## 1. Introduction

In early 2020, COVID-19 spread across the globe. Major events around the world have been postponed or canceled for fear of COVID-19. Given the characteristics of human-to-human transmission of COVID-19, some countries have proposed measures for combating COVID-19 using unmanned technologies [1, 2]. In addition to robotic buses, unmanned delivery vehicles and sweepers have also been installed. Some tasks involving frequent contact with people have been automated. The pandemic has also boosted the development of automation in different industries such as live streaming commerce and online shopping.

COVID-19 has also upended the traditional model of higher education. As campuses across the country sit empty in the midst of the pandemic, administrators are scrambling to prepare for what comes next. In many ways, the attack that COVID-19 represents on higher education is a straightforward assault on traditional views. This sudden pandemic has thrown the original teaching schedules of colleges into turmoil. With all education now being virtual,

online learning has become an opportunity to supply higher quality education. In the early pandemic, nearly all learning moved from the classroom to the Internet, which has accelerated the trend toward online learning and raised fundamental questions regarding online learning. What will be the lasting legacy of COVID-19 in higher education? Will the classroom that we once knew gradually return or could COVID-19 permanently transform how we learn?

In recent times, online learning models have been iterated and upgraded with the help of information technology [3–5]. There are three stages of online learning: video-on-demand- (VOD-) based online learning, live-streaming-based online learning, and real-time interactive online learning. In the late 1990s, online schools were distance education platforms with networks as a medium and videos were viewed through the Internet. This type of platform mainly relies on different forms of recording to record videos of lectures and upload them to the Internet. Students then follow what is happening in the lectures. The wave of online learning began in 2011 with live streaming. Teachers can interact with students to a certain degree

through live streaming and they can also answer student questions online. Live streaming online education has some advantages for online learning compared to the traditional PC era, which reduces the cost of learning to a significant extent and makes learning more fruitful. Live-streaming-based online learning is less engaging than real classes and it is easier to become distracted. However, learning is a real-time and two-way communication process and neither VOD nor live streaming learning can fully meet these two needs. Representing real-time communication scenarios between teachers and students is the biggest hurdle in online learning [6].

Interactive video live streaming is popular for online learning based on its interactivity and involvement, especially for large-scale online college learning in the post-pandemic era. However, given the limitations of data transmission and audio and video (AV) technology, it is difficult to avoid latency in interactive video live streaming, which negatively affects user experiences [7]. Artificial intelligence (AI) has significant promise for Internet applications and can serve as a positive force for the advancement of human society. By combining video live streaming with AI, it can be optimized in terms of coding and decoding (codec) technology, video transmission protocols, and other aspects to improve transmission efficiency and reduce latency to improve the performance of interactive video live streaming for large-scale online college learning.

Accordingly, the main contributions of this study can be summarized as follows. (i) An intraframe prediction optimization algorithm for the H.266 codec based on long short-term memory networks (LSTMs) is proposed. (ii) A real-time communication optimization based on Quick UDP Internet connections (QUIC) and the Luby Transform (LT) code video transmission protocol is proposed to improve the quality of interactive live video streaming.

The remainder of this paper is organized as follows. Section 2 reviews related work. In Section 3, the intraframe prediction optimization algorithm for the H.266 codec based on LSTMs is presented. In Section 4, real-time communication optimization based on the QUIC and LT code video transmission protocol is proposed. Experimental results are presented in Section 5. Section 6 concludes this paper.

## 2. Related Work

In recent years, many strategies for online learning have been proposed. In [8], the authors presented an enhanced recommendation method called adaptive recommendation based on an online learning style, which implements learning resource adaptation by mining learner behavioral data. In [9], the authors conducted in-depth exploration from the perspective of student self-efficacy by extending the four dimensions of online learning: a sense of effort, sense of control, sense of participation, and sense of environment. In [10], the authors attempted to provide an effective online teaching method and investigated the effects of online competency-based learning and design-based learning on enhancing student learning performance, self-directed learning readiness, and experience of online learning in an

online computing course. In [11], the authors examined the impact of personality traits, learning styles, gender, and online course factors (course difficulty, group affiliation, provided materials, etc.) on the academic success of students taking online courses and their overall success rate compared to traditional classes.

For large amounts of video data and high visual quality requirements, video codec technology has been rapidly developed and has become much more mature. In [12], a high-efficiency video codec-based spatial resolution scaling type for mixed resolution coding for frame interleaved multiview videos was proposed. In [13], the authors proposed a novel video transmission strategy that effectively transfers the computational complexity of video coding from the terminal to the cloud environment. In [14], a method for creating representative video sets covering all segments of user videos was proposed. In [15], novel compression tools for inter-/intraprediction, in-loop filtering, and entropy coding were proposed. In [16], the authors presented a model to distinguish regions of interest based on the VP9 video codec. Their model contained high motion-, contrast-, and color-sensitive areas that fit the human visual system. In [17], three new approaches for generating spatial intraprediction signals were supported: a line-wise application for conventional intraprediction modes coupled with a mode-dependent processing order, region-based template matching prediction method, and intraprediction modes based on neural networks. In [18], diamond adaptive rod pattern-search-algorithm-based block-matching motion estimation algorithms were proposed for multistranded codec hardware design to provide a high compression rate with less computational complexity. In [19], the authors proposed a deep-learning-based systematic approach that included an effective convolutional neural network structure, hierarchical training strategy, and video-codec-oriented switchable mechanism.

Several improved QUIC protocols for video transmission have also been investigated. For example, in [20], the authors proposed a novel design for the socket secure (SOCKS) protocol over QUIC (QSOCKS), which improved the browsing experience while enhancing reliability. In [21], the authors presented a security analysis of the QUIC handshake protocol based on symbolic model checking. In [22], the authors proposed a lightweight latency reduction scheme for the QUIC protocol. The proposed scheme calculated the average congestion window, which was utilized as the initial congestion window when a new connection was established. In [23], the authors proposed a modification of the handshaking mechanism for the QUIC protocol to minimize the overhead incurred by control signals and time required to update the congestion window size. In [24], the authors proposed QUIC-EST, which is a transmission scheme that combines the congestion control and multistream features of the recently proposed QUIC transport protocol with an optimized scheduling algorithm to maximize the value of information at the receiver. In [25], the authors proposed QUIC-based secure transport protocol to improve the transport performance of network traffic.

Additionally, many strategies for LT codes have been proposed. In [26], the authors proposed a second coding scheme based on LT codes under inactivation decoding. In [27], the authors proposed a modified version of LT encoding for the delivery phase to take advantage of channel coding. In [28], an improved LT code with a reverse coding framework was designed to reduce the error floor caused by low-degree information nodes.

### 3. Intraframe Prediction Optimization Algorithm for the H.266 Codec Based on LSTMs

In large-scale online college learning, the main traffic of interactive video live streaming is the same as that of traditional video live streaming. The signal source successively completes the information collection, encoding, and push streaming of streaming media while the terminals complete pull streaming, decoding, and playing.

Versatile video coding, which is also known as H.266, is a relatively new video encoding standard. Codecs are software packages that compress and decompress video files so they can use less space, which allows them to use fewer resources. H.266 can shrink files by up to 50% more than the current H.265 standard, which means that higher-definition streaming media information can be transmitted at a lower bandwidth. Therefore, the upgrading of codec technology can significantly reduce the cost of video storage and transmission, and decreased data transmission can reduce latency and improve overall stability. Therefore, students can receive better video quality under the same network conditions.

Intraframe prediction coding is the core of a video encoder. From H.265 to H.266, the main method for improving intraframe prediction performance is to add an elaborate angle direction scheme. However, there are some limitations to this method of relying solely on increasing the prediction direction. Typically, the brightness and chromaticity of two adjacent pixels are often relatively close in an

image, meaning color tends to change gradually. The goal of video coding is to use correlations to compress an image. The larger the prediction error, the more the bits required for coding and the less efficient the video compression.

LSTMs are designed for applications in which the input is an ordered sequence. LSTMs are a type of recurrent neural network (RNN), which are networks that reuse the output from a previous step as the input for the next step. Similar to all neural networks, nodes perform calculations on inputs and return output values. In a recurrent network, this output is then used in combination with the next element as the input for the next step. In an LSTM, nodes are recurrent, but they also have an internal state. Each node uses its internal state as a working memory space, meaning that information can be stored and retrieved. The input value from the previous output and the internal state are both used in node calculations. The results of calculations are used not only to provide output values, but also to update the state. Like any neural network, LSTM nodes have parameters that determine how inputs are used in calculations and they are also known as gates that control the flow of information within nodes.

LSTMs solve the problems of vanishing gradients and gradient explosions caused by RNN networks during training, so they can make better use of the spatial correlations between adjacent pixels to express brightness changes in the prediction direction. Based on our analysis of the variation rules of pixel brightness in the prediction direction, this paper proposes an angle prediction and compensation algorithm based on LSTMs that makes a secondary prediction for prediction errors in the horizontal and vertical angle models of H.266 and aids the standard linear prediction model to improve prediction accuracy.

3.1. *Horizontal and Vertical Mode of H.266.* The typical equation for H.266 angle prediction is defined as follows:

$$\text{pre}[x, y] = \frac{\{(32 - \omega LI) * \text{ref}[x + iPM + 1] + \omega LI * \text{ref}[x + iPM + 2] + 16\}}{32}, \quad (1)$$

where  $\omega LI$  is the linear interpolation weight,  $iPM$  is the index of the prediction model,  $\text{pre}[x, y]$  is the predicted value of a pixel, and  $\text{ref}[x]$  is the one-dimensional reference pixel set projected according to the prediction direction.

The horizontal and vertical modes of the angle prediction have typical characteristics. During the prediction process, the predicted values of each row or column are equal to the values of the same reference pixels. When the prediction mode is vertical, the predicted value of the pixel is equal to the value of the reference pixel above the current pixel, which can be defined as follows:

$$\text{pre}[x, y] = \text{ref}[x + 1]. \quad (2)$$

When the prediction mode is horizontal, the predicted value of the pixel is equal to the value of the reference pixel to the left of the current pixel and  $x$  on the right side of (2) can be directly replaced with  $y$ .

We assume that the prediction mode of the current pixel block is horizontal and the width and height of the pixel block are  $w$  and  $h$ , respectively. The  $x$ -th reference pixel above is denoted by  $abo[x]$ , the  $y$ -th reference pixel on the left is denoted by  $\text{lef}[y]$ , the pixel value of the  $x$ -th column in the  $y$ -th row of the current pixel block is denoted by  $p[x, y]$ , and the predicted value  $\text{pre}[x, y] = \text{lef}[y]$ . The prediction errors  $\text{pe}[x, y]$  can be calculated as follows:

$$pe[x, y] = p[x, y] - pre[x, y] = p[x, y] - lef[y]. \quad (3)$$

Given the definitions above, in the horizontal and vertical modes of the H.266 intraframe prediction scheme, the prediction results are directly obtained from the left or above reference pixels. In this method, for coding blocks with complex internal details, the prediction errors are too large and the coding efficiency is reduced. To solve this problem, this study leveraged the ability of LSTMs of expressing sequence data and making a secondary prediction of prediction errors to compensate for the deficiencies of the standard linear prediction model. The core concept of the proposed algorithm is that the prediction error data of the horizontal and vertical modes in the output results of the standard encoder are counted; then the error data are classified according to the size of the coding block, and then the LSTM model is trained. The LSTM model is used to calculate the prediction compensation values in real time and compensate for the original prediction data to improve prediction accuracy.

**3.2. Network Architecture.** The number of neurons in the input layer of the network is twice the predicted block width (i.e.,  $2 \times w$ ) and the number of neurons in the output layer is equal to the predicted block width (i.e.,  $w$ ). The network depth is  $K$  and the time step is the height of the coding block (i.e.,  $h$ ).

In the network architecture of this algorithm, the construction method for the input data is one of the keys to the algorithm. Here, the construction method for the input data is introduced by considering the horizontal prediction mode as an example. In the horizontal prediction mode, the prediction error of each row is the difference between the pixel values of each pixel of the row and the left reference pixel. The reference row above the prediction block is also calculated to obtain  $h + 1$  rows of prediction errors.

According to the spatial correlation of images and how the horizontal prediction model works, the prediction errors of adjacent rows are also correlated. To improve the prediction effect, for each row to be predicted, the proposed method selects the prediction errors ( $pe_{n-1}$  and  $pe_{n-2}$ ) of two rows above the current row as the inputs for the network. The vertical prediction mode is treated similarly to the horizontal prediction mode and is not described here.

According to the characteristics of error data, the tanh function was selected as the activation function for the input layer in this study and the prediction results were obtained through the hidden layer and output layer. The tanh function is also known as the hyperbolic tangent activation function. The mean squared error (MSE) function was used as the loss function during network training as follows:

$$MSE = \frac{\sum_{i=0}^{w-1} (pre[i] - y[i])^2}{w}. \quad (4)$$

## 4. Real-Time Communication Optimization Based on QUIC and the LT Codes Video Transmission Protocol

The current mainstream Internet video transmission protocols are the real-time messaging protocol and hypertext transfer protocol, both of which are based on the transmission control protocol (TCP). The TCP is designed to be complex to provide a reliable transmission service, which increases the latency of data packets and cannot meet the real-time requirements of interactive live video streaming. In contrast, the user datagram protocol (UDP) has a high transmission efficiency and low latency. However, as a result of its unreliability, even if it is directly applied to interactive live videos on the Internet, it cannot meet the requirements for the smooth transmission of streaming media information. Currently, there are many optimization schemes for UDP, including the reliable UDP model and QUIC developed by Google. QUIC is a UDP-based low latency Internet transmission protocol. According to research results, the lag rate of QUIC pull streaming is 57% lower than that of TCP pull streaming. Although the QUIC protocol has the characteristics of zero round-trip time (RTT), low latency, connection migration, and security, it can still be optimized and improved according to the requirements of real-time communication scenarios. We attempted to improve the QUIC protocol for real-time communication. Our improvements include improving the corresponding format of the transmission frame and partial reliability marker, which makes the QUIC more appropriate for a communication scenario that allows moderate packet loss. The improved QUIC protocol is called rt-QUIC, where “rt” stands for “real-time.”

**4.1. Designing an Improved Frame Format.** The packet format of rt-QUIC must transmit a parameter with a maximum frame size of  $\delta$  and this parameter is used before data stream exchanges at multiple communication ends. In addition to informing each other  $\delta$ , the communication ends also need to carry a series of parameters, including the frame length, load, and data type that the communication end can receive. When rt-QUIC is used to send “unreliable” packets between multiple communication ends, it generates a new frame, which is sent immediately and may be merged with other frames during transmission. rt-QUIC will do its best to deliver these unreliable merged frames. When the communication end determines that the transmitted data at one end may be lost, it directly informs the end that a frame has been lost. The rt-QUIC packet will incur acknowledgment (ACK) frame latency when a packet is lost and it will not retransmit the packet after the frame is lost but will rearrange the serial number of the packets. The response receiving time is controlled by the response time of the ACK frames and the lost data frames can be received and acknowledged at a later time. To prevent the loss of the tail of the data stream, a reliability mark is added to the end of the stream. When a data stream that the communication end attempts to use has been lost, it will receive a stream of size

zero, which is a supplementary stream to fill in the lost data. Finally, the last byte in each stream is reliably transmitted and the data stream can be delivered as completely as possible.

*4.2. Partial Reliability Marker.* The design of a partial reliability marker is suitable for AV transmission scenarios. Each frame is marked and only one bit is used to indicate whether it is reliable. Adding an identifier bit may slightly increase the cost of sending and receiving, but this is largely inconsequential. This is mainly because AV streaming media files are often accompanied by codecs in the transmission process. As shown in Figure 1, in the codec process of AV transmission, the AV stream of the sender is first coded and then the coded data are transmitted. In the coding process, the partial reliability bit will not increase the cost of the sender. For the receiver, the file encoded by the sender is transmitted through a peer-to-peer or server connection and the power of the receiving end must be used for decoding. The decoding process rarely causes an increase in processing time. Because the receiving end must judge the reliability marker in a packet and process data streams in different scenarios, overall, the communication service experience is improved.

*4.3. LT Codes.* In addition to allowing moderate packet loss in the network layer, fountain code technology can be used to realize data integrity under the condition of a small amount of packet loss. Fountain codes are approximations of an ideal digital fountain. A digital fountain allows a client to obtain droplets or encoded message packets from a server and use them to rebuild an encoded file. The code makes it so that the actual packets are received and the order in which they are received does not matter. An ideal digital fountain can generate an infinite supply of droplets from the original data and a receiver can reconstruct an original data file made of  $N$  packets once  $N$  droplets have been received. Digital fountains are similar to water fountains in that we do not care which drops we get from the fountain as long as we get sufficient drops to fill our water bottle. In 1998, Luby invented the LT code. LT codes rely on an exclusive OR operation to encode and decode a message. LT codes are called rate lists because their encoding algorithm can produce an infinite supply of message droplets in principle. Fountain codes are useful in scenarios where packet loss is likely and in scenarios where a receiver cannot communicate with a sender.

The interactive video live streaming of online college learning considered in this study is based on LT codes combined with rt-QUIC (rt-QUIC-LT), with a sender and receiver. The sender can be a client or server. Likewise, the receiver can also be a client or server. In this study, we assumed that the client is the sender and the server is the receiver. First, the sender loads a file and divides it into data blocks of the same size. The sender then divides each data block into  $k$  source characters for LT coding. Finally, the encoded characters and encoded information are packaged and sent to the receiver. After receiving sufficient encoded

packets, the receiver unpacks the encoded characters and encoded information, and then LT decoding is performed. After decoding, the transmitted video stream can be recovered.

*4.4. Implementation of the Receiver.* Online college learning realizes interactive video live streaming based on the rt-QUIC socket, which is based on the connectionless UDP. Therefore, interactive video live streaming between a sender and receiver does not establish a direct connection. The rt-QUIC video transmission scheme has high real-time performance and a fast transmission rate without data confirmation and retransmission mechanisms. Interactive video live streaming for online college learning uses a connectionless socket on the receiver side to communicate with the sender through the following six steps:

Step 1: The receiver first creates and instantiates a socket based on the port number and IP address. The port number was set to 8735 in this study.

Step 2: Each frame of video streaming is added with a partial reliability marker.

Step 3: The Bind() function is called to bind the local address and port.

Step 4: The receiver initializes a receiving thread and waits for data to be received.

Step 5: Interactive video live streaming uses the recvfrom() function to receive data from the sender.

Step 6: The socket is closed and all resources are released.

According to these steps, a flow chart for creating a socket on the receiver side is presented in Figure 2.

*4.5. Implementation of the Sender.* First, a receiver creates a socket and binds the address to the port of the sender and then creates a receiving thread and waits for the sender to send data. In the meantime, the sender creates a socket and a sending thread, which begins sending data to the specified address until all data have been sent and no connection is established with the receiver. When the sender begins to send data, the receiver begins to receive data and obtains information regarding the transferred video stream from the received data until all data are received.

In interactive video live streaming for online college learning through the sockets of rt-QUIC, LT codes are used to ensure the reliability of video streaming. First, LT encodes a video stream to generate sufficient encoded characters at the sender. The encoded characters and video stream are then packaged and sent to the receiver through sockets. After receiving the data, the receiver unpacks the encoded characters and recovers the video stream using the LT decoder. LT codes can ensure the reliability of video streaming transfer, which is determined by the LT code characteristics. The nonbit-rate characteristic of LT codes enables a sender to generate an infinite number of encoded characters and a receiver to recover a video stream after receiving sufficient

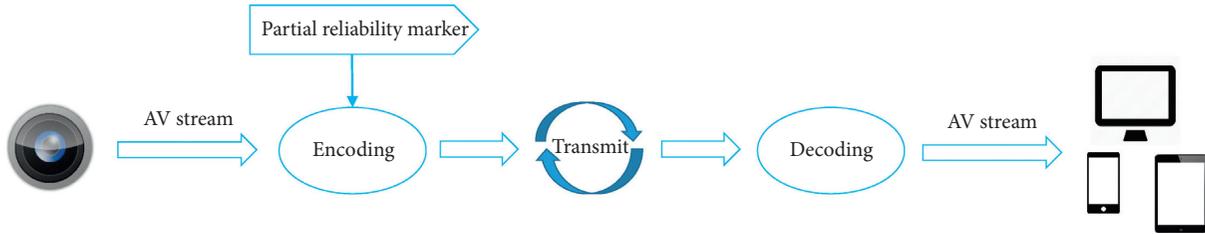


FIGURE 1: AV streaming codec process.

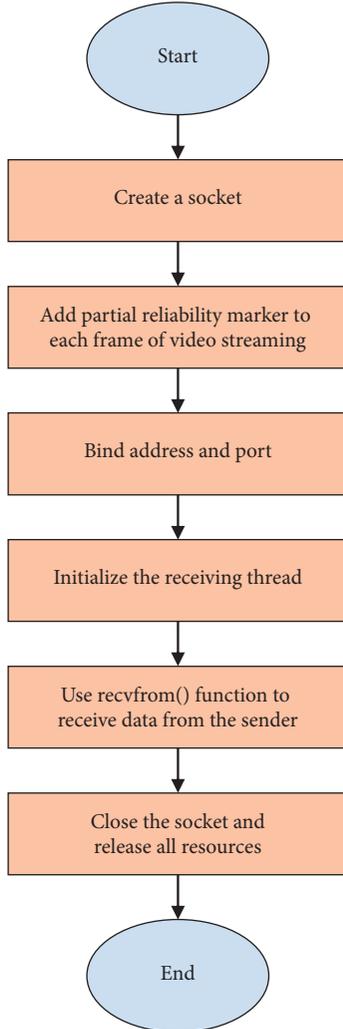


FIGURE 2: Socket of receiver.

encoded characters. Data lost during video streaming transfer and the sequence of received data do not affect video stream recovery.

## 5. Experiments and Results Analysis

**5.1. Setup.** In this study, the encoding, transmission protocol, and transmission efficiency of traffic in interactive video live streaming for online college learning were optimized. Simulations were executed on a computer with an Intel i9-11900KF 3.7 GHz CPU, 32 GB of RAM (3333 MHz),

and NVIDIA RTX 3080 Ti, 12GB GDDR6X GPU. To verify the efficiency of the intraprediction optimization algorithm proposed in this paper, the test platform VTM-7.3 for H.266 was used as a test environment. The encoded video stream quality was measured using the peak signal-to-noise ratio (PSNR) and Bjontegaard delta bit rate (BDBR) [29]. Additionally, we compare the proposed rt-QUIC and LT code video transmission to three benchmarks: QUIC, QUIC-EST, and QSOCKS.

### 5.2. Parameter Settings

**5.2.1. Depth of the Network.** The size and depth of the network have a significant impact on network performance. Network performance can be effectively improved by increasing network depth. However, not all types of networks can achieve higher performance by using deeper layers. Based on the intraframe prediction data of live video streaming coding, this study compared network performance with different sizes and depths, and LSTM loss function curves for the same size and different depths were obtained. As shown in Figure 3, the model with six hidden layers can achieve the lowest training cost with an increasing number of training epochs. Therefore, this study adopted LSTMs with six hidden layers for training.

**5.2.2. Size of Hidden Layers.** The size of the input layer of the network was fixed at  $2 \times w$  and the size of the output layer was fixed at  $w$ . The size of the hidden layer could be adjusted. In [30], a clustering strategy was used to divide a population into several clusters. In this study, network performance with different hidden layer sizes was calculated, as shown in Figure 4. One can see that the network performance improves with a larger size of hidden layer, but this change becomes less obvious when the size is above 48. Therefore, the size of the hidden layer selected in this study was 48.

**5.3. Comparative Analysis.** Simulations were performed to examine video streaming quality, video streaming latency, average throughput, and online learning quality of experience (QoE). The number of simulations was 2000.

**5.3.1. Video Streaming Quality.** PSNR is directly proportional to video streaming quality, but it is often impossible to consider both a low bitrate and a high recovery quality simultaneously when measuring the quality of video encoded by the proposed algorithm. Therefore, the change in bitrate

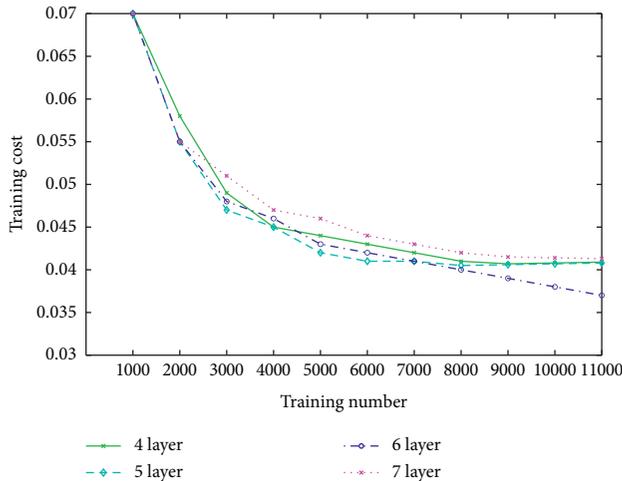


FIGURE 3: Cost curves of networks with the same size and different depths.

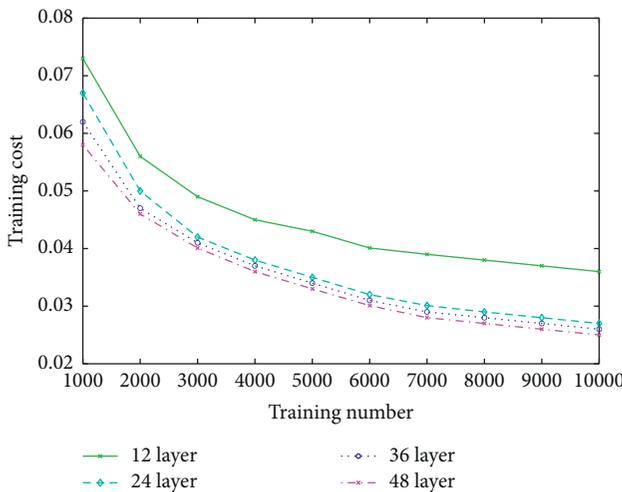


FIGURE 4: Cost curves of networks with different hidden layer sizes.

after coding was evaluated on the premise of restoring the same video image quality. The PSNR and BDBR of the proposed algorithm and benchmarks with different sequences are listed in Table 1.

In Table 1, one can see that the PSNR of the proposed algorithm is smaller than those of the two benchmarks for different sequences, indicating that the proposed algorithm performs well in terms of video streaming quality. In terms of BDBR, the average bitrate change is less than 1%, meaning that it takes less time to restore the same video streaming quality. For the horizontal and vertical prediction angle models, the LSTM is used to perform secondary prediction, which compensates for the original intraframe prediction results and effectively reduces prediction errors.

**5.3.2. Video Streaming Latency.** The latency test results under the conditions of network transmission are presented in Table 2. The simulations were driven by video streaming

data from the H.266 codec based on LSTMs with resolutions of 240p, 360p, 480p, 720p, 1080p, 1440p, and 2160p.

As shown in Table 2, the transmission latency of the proposed algorithm is lower than those of the two benchmarks before and after decoding. LSTMs solve problems that RNNs cannot handle in terms of long-term dependencies, which is important to this study’s large-scale online learning scenario. The hidden layer in a typical RNN has only one state, which is sensitive to short-term inputs. LSTMs add an additional state to hold long-term information. The tanh activation function is used to help regulate video streams flowing through the network.

**5.3.3. Average Throughput.** The proposed rt-QUIC-LT was compared to three benchmarks. The experimental variables were the RTT and bandwidth. After several rounds of testing, the throughput of the results in each round was averaged as a throughput reference value. In Figure 5, one can see that there are no significant differences between the performances of the four protocols when the RTT is small and the throughput can basically remain near the peak value. However, the throughputs of the three benchmarks decrease with an increase in the RTT. Therefore, it can be inferred that the processing capacity of the three benchmarks is not as high as that of rt-QUIC-LT when network congestion occurs. Because rt-QUIC-LT improves the frame format for real-time communication scenarios, it can be combined with the selective send strategy to achieve smooth and reliable performance in poor network conditions.

When the variable is bandwidth, the experimental performance results are presented in Figure 6, where the average throughput of the four protocols increases with an increase in bandwidth. However, when comparing the four protocols, one can see that rt-QUIC-LT yields its best performance under a given bandwidth more quickly, whereas the three benchmarks reach the optimal state more slowly. It can be inferred that the improvement of the frame format and packet loss strategy plays an important role in rt-QUIC-LT. The rt-QUIC-LT protocol actively loses some retransmitted and meaningless packets, and the transmission cost is relatively low. Therefore, the average throughput of the rt-QUIC-LT protocol is better than those of the benchmarks under the same bandwidth.

Figure 7 presents performance comparisons of the four protocols in real-time communication scenarios. The preset packet loss rate of the test network environment is 1% and the preset latency is 50 ms. According to the simulation results, QUIC-EST and QSOCKS are the first to reach the maximum throughput in the initial stage of connection establishment, but they do not maintain the maximum throughput steadily. In contrast, the average throughput of rt-QUIC-LT shows a steady increase with less fluctuation. Overtime, QUIC-EST and QSOCKS exhibit several periods of lag, leading to low performance in terms of effective throughput. Although the rt-QUIC-LT protocol gives up partial reliability, its stability is enhanced and transmission performance can be maintained at a higher level. LT codes

TABLE 1: The PSNR and BDBR of the proposed algorithm and benchmarks with different sequences.

Test sequence	fCU-BIDBF		fISP-CU		This paper	
	BDBR (%)	PSNR (dB)	BDBR (%)	PSNR (dB)	BDBR (%)	PSNR (dB)
Cactus	1.48	-0.06	2.56	-0.11	0.66	-0.02
Foodmarket	1.21	-0.03	2.71	-0.08	1.11	-0.04
Catrobot	2.08	-0.13	2.65	-0.15	0.66	-0.08
Arena of Valor	2.46	-0.12	2.38	-0.13	0.32	-0.04
Slide Editing	1.04	-0.08	2.59	-0.09	0.25	-0.05
BQ Mall	1.54	-0.11	1.37	-0.16	0.44	-0.04
Fourpeople	1.99	-0.13	3.69	-0.21	1.17	-0.03
Johnny	1.37	-0.09	3.72	-0.16	1.33	-0.02
Average	1.65	-0.09	2.71	-0.14	0.74	-0.04

TABLE 2: The comparison of network transmission latency under different resolutions.

Resolution	fCU-BIDBF			fISP-CU			This paper		
	Before encoding (ms)	After decoding (ms)	Time difference (ms)	Before encoding (ms)	After decoding (ms)	Time difference (ms)	Before encoding (ms)	After decoding (ms)	Time difference (ms)
240p	11.05	11.36	0.31	10.54	10.79	0.25	9.81	9.89	0.08
360p	15.17	18.99	3.82	13.38	15.89	2.51	12.25	13.01	0.76
480p	20.96	31.49	10.53	18.94	23.57	4.63	15.67	16.78	1.11
720p	30.45	45.68	15.23	24.56	30.02	5.46	20.08	22.48	2.4
1080p	56.37	75.76	19.39	50.19	58.46	8.27	44.52	47.08	2.56
1440p	180.31	204.34	24.03	120.48	131.64	11.16	100.69	103.96	3.27
2160p	311.72	343.66	31.94	260.49	281.33	20.84	210.77	214.12	3.35
Average	89.43	104.47	15.04	71.23	78.81	7.59	59.11	61.05	1.93

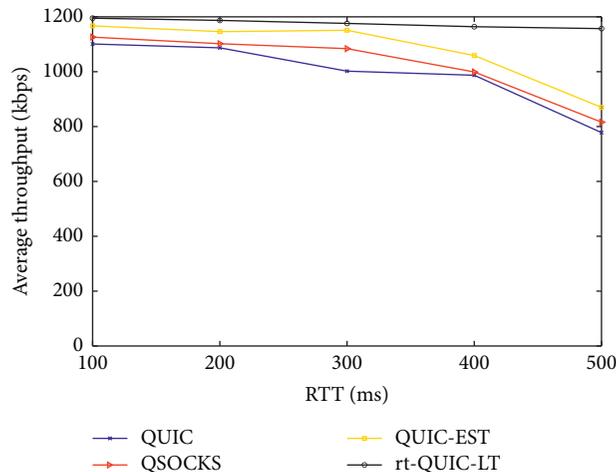


FIGURE 5: Average throughput of different RTT.

inherit the advantages of fountain code while significantly reducing the complexity of coding and decoding.

The performances of the four protocols in a poor network scenario are now compared. Figure 8 presents a comparison of the results when the network packet loss rate is 10% and the network latency is 250 ms in the test environment. Based on this poor simulated network environment, the results reveal that QUIC-EST and QSOCKS suffer from multiple transmission lag events. Because they are affected by packet loss, their effective throughput fluctuates significantly, resulting in a poor communication experience.

In contrast, rt-QUIC-LT has an advantage in this scenario. rt-QUIC-LT takes advantage of its incomplete reliability and drops packets that do not need to be retransmitted to reduce fluctuations in network quality. The improved rt-QUIC-LT can adapt to fluctuations in the network state. Therefore, in the case of poor network conditions, although lag still occurs, the effective throughput does not decrease significantly and the entire communication experience is acceptable. Finally, it can be concluded that rt-QUIC-LT is suitable for real-time AV communication and video streaming transmission scenarios.

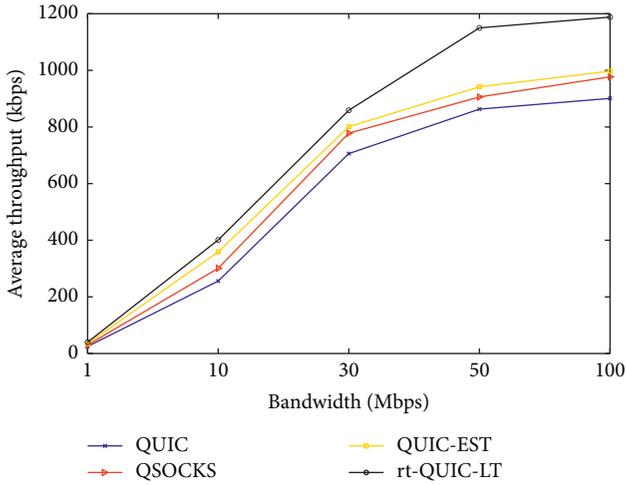


FIGURE 6: Average throughput of different bandwidths.

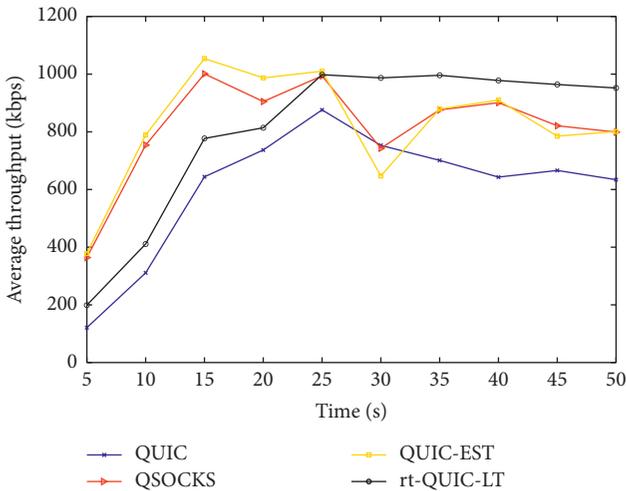


FIGURE 7: Average throughput under 1% packet loss and 50 ms latency.

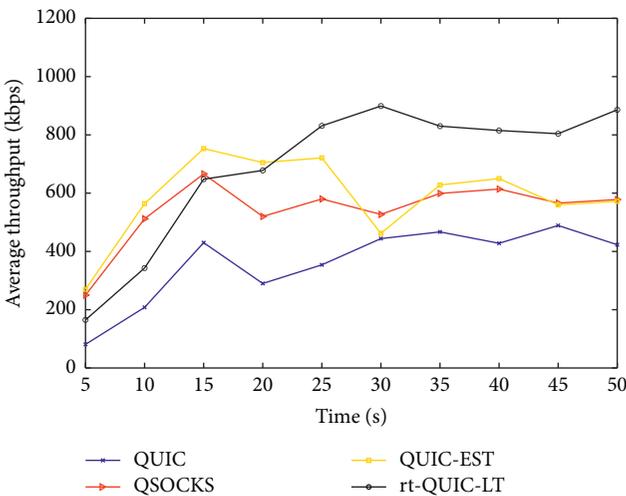


FIGURE 8: Average throughput under 10% packet loss and 250 ms latency.

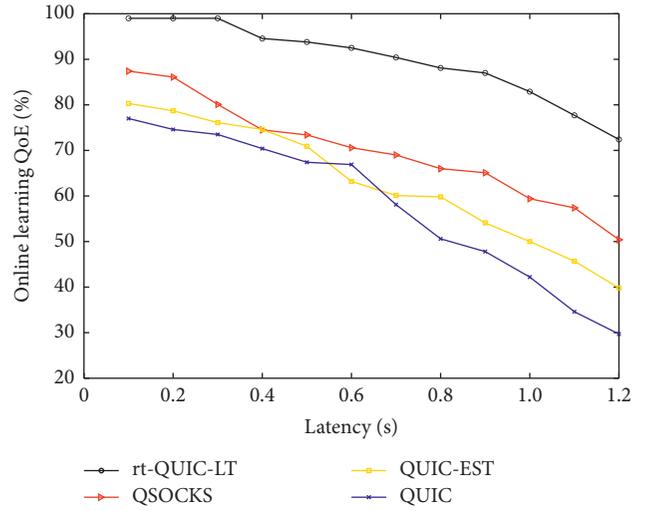


FIGURE 9: Online learning QoE.

5.3.4. *Online Learning QoE.* Figure 9 presents the QoE of large-scale online college learning under different network transmission latencies. As shown in Figure 9, rt-QUIC-LT can guarantee better online learning QoE under low latency. However, the online learning QoE decreases with an increase in latency. Even though the online learning QoE of rt-QUIC-LT decreases, it is still higher than those of the three benchmarks. The online learning QoE of the three benchmarks decreases more rapidly at higher latency, which verifies that the real-time communication optimization based on the QUIC and LT code video transmission protocol proposed in this paper can effectively improve the online learning QoE.

## 6. Conclusions

Interactive video live streaming plays a significant role in large-scale online college learning in the postpandemic era. Regarding video codecs, we proposed an intraframe prediction optimization algorithm for the H.266 codec based on LSTMs. Additionally, real-time communication optimization based on the QUIC and LT code video transmission protocol was proposed to improve the quality of interactive video live streaming. Experimental results demonstrated that the proposed algorithms perform well in terms of video streaming quality, video streaming latency, and average throughput.

The quality of real-time AV transmission is very strict and the realization of online real-time interactive scenes faces four main challenges: high concurrency, high reliability, low latency, and antiweak network characteristics. In addition to low latency, these other three aspects will be studied in the future. We will also enhance the experience of online learning by studying virtual reality and augmented reality.

## Data Availability

All the data used to support the findings of the study are included within the article.

## Conflicts of Interest

The author declares no conflicts of interest in this paper.

## References

- [1] O. Ozkan and O. Atli, "Transporting COVID-19 testing specimens by routing unmanned aerial vehicles with range and payload constraints: the case of Istanbul," *Transportation Letters-The International Journal of Transportation Research*, vol. 13, no. 5-6, pp. 4482-4491, 2021.
- [2] S. Manigandan, P. K. Thaloor Ramesh, N. T. L. Chi, and K. Brindhadevi, "Early detection of SARS-CoV-2 without human intervention to combat COVID-19 using drone technology," *Aircraft Engineering & Aerospace Technology*, vol. 93, no. 1, pp. 85-88, 2021.
- [3] D. Shen, M.-H. Cho, C.-L. Tsai, and R. Marra, "Unpacking online learning experiences: online learning self-efficacy and learning satisfaction," *The Internet and Higher Education*, vol. 19, pp. 10-17, 2013.
- [4] W. Holmes, Q. Nguyen, J. Zhang, M. Mavrikis, and B. Rienties, "Learning analytics for learning design in online distance learning," *Distance Education*, vol. 40, no. 3, pp. 309-329, 2019.
- [5] S. R. Palmer and D. M. Holt, "Examining student satisfaction with wholly online learning," *Journal of Computer Assisted Learning*, vol. 25, no. 2, pp. 101-113, 2009.
- [6] A. W. Cole, L. Lennon, and N. L. Weber, "Student perceptions of online active learning practices and online learning climate predict online course engagement," *Interactive Learning Environments*, vol. 29, no. 5, pp. 866-880, 2021.
- [7] T. Zhao, Q. Liu, and C. W. Chen, "QoE in video transmission: a user experience-driven strategy," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 1, pp. 285-302, 2017.
- [8] H. Chen, C. Yin, R. Li, W. Rong, Z. Xiong, and B. David, "Enhanced learning resource recommendation based on online learning style model," *Tsinghua Science and Technology*, vol. 25, no. 3, pp. 348-356, 2020.
- [9] H. Liu, Y. Ye, and H. Jiang, "Self-efficacy in home-based online learning environments," *Journal of Internet Technology*, vol. 22, no. 3, pp. 557-567, 2021.
- [10] C.-W. Tsai, "Applying online competency-based learning and design-based learning to enhance the development of students' skills in using PowerPoint and Word, self-directed learning readiness, and experience of online learning," *Universal Access in the Information Society*, vol. 19, no. 2, pp. 283-294, 2020.
- [11] E. Idrizi, S. Filiposka, and V. Trajkovijk, "Analysis of success indicators in online learning," *International Review of Research in Open and Distance Learning*, vol. 22, no. 2, pp. 205-223, 2021.
- [12] B. Mallik, A. Sheikh-Akbari, and A.-L. Kor, "Mixed-resolution HEVC based multiview video codec for low bitrate transmission," *Multimedia Tools and Applications*, vol. 78, no. 6, pp. 6701-6720, 2019.
- [13] T. C.-W. Lei and F.-S. Tseng, "Light-weight video codec at terminal for video coding in the cloud," *Journal of Signal Processing Systems*, vol. 91, no. 6, pp. 627-639, 2019.
- [14] A. V. Zvezdakova, D. L. Kulikov, S. V. Zvezdakov, and D. S. Vatolin, "BSQ-rate: a new approach for video-codec performance comparison and drawbacks of current solutions," *Programming and Computer Software*, vol. 46, no. 3, pp. 183-194, 2020.
- [15] K. Choi, J. Chen, M. W. Park et al., "Video codec using flexible block partitioning and advanced prediction, transform and loop filtering technologies," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 5, pp. 1326-1345, 2020.
- [16] Z. Zhang, T. Jing, and C. Tian, "Hybrid context modeling and wavelet transform for region of interest detection and rate control based on VP9 video codec," *Concurrency and Computation: Practice and Experience*, vol. 31, no. 10, 2019.
- [17] J. Pfaff, H. Schwarz, D. Marpe et al., "Video compression using generalized binary partitioning, trellis coded quantization, perceptually optimized encoding, and advanced prediction and transform coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 5, pp. 1281-1295, 2020.
- [18] S. M. Balamurugan and R. Seshasayanan, "An efficient field-programmable gate array-based hardware oriented block motion estimation algorithm based on diamond adaptive rood pattern search algorithm for multi-standard video codec," *Transactions of the Institute of Measurement and Control*, 2021.
- [19] D. Ding, L. Kong, and G. Chen, "A switchable deep learning approach for in-loop filtering in video coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 7, pp. 1871-1887, 2020.
- [20] M. R. Kanagarathinam, S. Singh, S. R. Jayaseelan, M. K. Maheshwari, G. K. Choudhary, and G. Sinha, "QSOCKS: 0-RTT proxiification design of SOCKS protocol for QUIC," *IEEE Access*, vol. 8, pp. 145862-145870, 2020.
- [21] J. Zhang, L. Yang, X. Gao, G. Tang, J. Zhang, and Q. Wang, "Formal analysis of QUIC handshake protocol using symbolic model checking," *IEEE Access*, vol. 9, pp. 14836-14848, 2021.
- [22] J. Jung and D. An, "Access latency reduction in the QUIC protocol based on communication history," *Electronics*, vol. 8, no. 10, 2019.
- [23] P. Kharat and M. Kulkarni, "Modified QUIC protocol with congestion control for improved network performance," *IET Communications*, vol. 15, no. 9, pp. 1210-1222, 2021.
- [24] F. Chiariotti, A. A. Deshpande, M. Giordani, K. Antonakoglou, T. Mahmoodi, and A. Zanella, "QUIC-EST: a QUIC-enabled scheduling and transmission scheme to maximize VoI with correlated data flows," *IEEE Communications Magazine*, vol. 59, no. 4, pp. 30-36, 2021.
- [25] J. Zhang, X. Gao, L. Yang, T. Feng, D. Li, and Q. Wang, "A systematic approach to formal analysis of QUIC handshake protocol using symbolic model checking," *Security and Communication Networks*, vol. 2021, Article ID 1630223, 12 pages, 2021.
- [26] A. Severinson, A. Graell Amat, and E. Rosnes, "Block-diagonal and LT codes for distributed computing with straggling servers," *IEEE Transactions on Communications*, vol. 67, no. 3, pp. 1739-1753, 2019.
- [27] A. H. F. Raouf, J. Abouei, and M. Jaseemuddin, "Delivery phase in cache-based wireless networks with modified LT codes," *Physical Communication*, vol. 42, 2020.
- [28] X. Song, N. Cheng, Y. Liao, S. Ni, and T. Lei, "Design and analysis of LT codes with a reverse coding framework," *IEEE Access*, vol. 9, pp. 116552-116563, 2021.
- [29] B. Abdallah, F. Belghith, M. A. Ben Ayed, and N. Masmoudi, "Low-complexity QTMT partition based on deep neural network for Versatile Video Coding," *Signal, Image and Video Processing*, vol. 15, no. 6, pp. 1153-1160, 2021.
- [30] L. Ma, S. Cheng, and Y. Shi, "Enhancing learning efficiency of brain storm optimization via orthogonal learning design," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 51, no. 11, pp. 6723-6742, 2021.