

Vol 2 Issue 02 (2026)

<http://fsrma.org.hostajournal.com/>

ONLINE ISSN: 3134-7517

PRINT ISSN: 3134-7509

STQ-Scheduler: A Secure and Throughput-Aware Deep Reinforcement Learning Framework for QoE-Driven Resource Scheduling in Distributed Video Streaming Systems

Yi-Chun Chang

University of California, Berkeley, Berkeley, CA, USA
ycchang@berkeley.edu

Minwei Jiang

Cornell University, Ithaca, NY, USA
mj422@cornell.edu

Abstract.

With the rapid growth of large-scale video streaming services, cloud gaming, and edge-assisted media delivery, ensuring high Quality of Experience (QoE) under dynamic network conditions and heterogeneous edge infrastructures has become a critical challenge. In practical systems, issues such as burst traffic, cross-edge latency variability, and noisy or unsafe data streams often lead to suboptimal resource utilization and degraded user experience. Meanwhile, conventional rule-based schedulers and static data processing pipelines are unable to jointly address the challenges of efficient model training and adaptive resource allocation. To tackle these problems, this paper proposes STQ-Scheduler, a secure and throughput-aware deep reinforcement learning framework that integrates high-throughput data processing, Transformer-based QoE prediction, and Proximal Policy Optimization (PPO)-based resource scheduling. The framework incorporates real-time data cleaning, anomaly filtering, and online feature transformation to ensure data quality and prevent data processing from becoming a bottleneck in distributed training. Furthermore, a multi-objective reward function is designed to jointly optimize QoE, latency, throughput, and system cost, enabling adaptive scheduling decisions across distributed edge nodes. Experimental results demonstrate that STQ-Scheduler significantly outperforms baseline methods in distributed video streaming environments. Specifically, it reduces average latency from 215 ms to 162 ms and improves throughput from 910 req/s to 1075 req/s under normal workloads. Under burst conditions, it maintains over 1020 req/s while reducing SLA violation rates from 10.5% to 3.9%, confirming its robustness and effectiveness in QoE-driven resource scheduling.

Keywords:

Deep Reinforcement Learning; Video Streaming; QoE Optimization; Distributed Training; Edge Computing; Resource Scheduling; Data Cleaning; Transformer

1. INTRODUCTION

With the rapid expansion of video streaming services, cloud gaming, and real-time multimedia applications, ensuring high Quality of Experience (QoE) has become a critical requirement for next-generation distributed systems. Edge computing has emerged as a key paradigm to reduce

latency and improve service responsiveness by deploying computational resources closer to end users. However, in practical video streaming scenarios, heterogeneous edge nodes, dynamic network conditions, bursty traffic patterns, and cross-node latency variations pose significant challenges to efficient resource scheduling. In addition, large-scale data generated by streaming systems often contain noise, anomalies, and potential security risks, making real-time data processing and model training increasingly complex. Traditional scheduling approaches, such as rule-based load balancing and threshold-driven auto-scaling, primarily focus on system-level metrics like CPU utilization or request counts, while overlooking user-centric QoE indicators such as startup delay and rebuffering events. Moreover, existing distributed training pipelines often suffer from low throughput due to inefficient data preprocessing, which becomes a bottleneck when integrating data-driven models into real-time scheduling systems. These limitations highlight the need for a unified framework that can jointly address secure data processing, efficient model training, and adaptive resource scheduling. To tackle these challenges, this paper proposes STQ-Scheduler, a secure and throughput-aware deep reinforcement learning framework for QoE-driven resource scheduling in distributed video streaming systems. The proposed framework integrates a high-throughput data pipeline for real-time data cleaning and feature transformation, a Transformer-based QoE prediction model to capture temporal dependencies in user behavior and network conditions, and a Proximal Policy Optimization (PPO)-based scheduler to dynamically allocate resources across edge nodes. By incorporating QoE predictions into the scheduling decision process, STQ-Scheduler enables a shift from traditional system-centric optimization to user-centric service optimization.

The main contributions of this work are summarized as follows:

- (1) We propose a unified framework that integrates secure data processing, high-throughput distributed training, and QoE-driven resource scheduling for video streaming systems.
- (2) We design a Transformer-based QoE prediction model that effectively captures temporal and contextual dependencies in streaming workloads.
- (3) We develop a PPO-based adaptive scheduling mechanism that dynamically optimizes resource allocation across heterogeneous edge nodes.
- (4) We construct an end-to-end closed-loop optimization system that continuously improves scheduling performance through feedback-driven learning.

Overall, this work provides a practical and scalable solution for intelligent resource management in modern distributed video streaming environments.

2. Related Work

In recent years, the rapid growth of video streaming services and edge computing has driven extensive research on QoE-aware optimization, distributed training systems, and intelligent resource scheduling. This section reviews the most relevant studies from three perspectives: QoE-driven video streaming optimization, high-throughput distributed training and data processing, and deep learning-based resource scheduling.

2.1 QoE-Driven Video Streaming Optimization

QoE optimization has become a central topic in video streaming systems, as user experience directly affects service retention and engagement. Early works focused on adaptive bitrate (ABR) algorithms, such as Pensieve [1], which applies reinforcement learning to dynamically adjust video bitrate selection based on network conditions. Similarly, Yin et al. [2] proposed a control-theoretic approach for ABR streaming, emphasizing stability and fairness. Mao et al. [3] further extended RL-based approaches for network resource management, demonstrating improved adaptability under varying conditions. Recent studies have explored the relationship between QoE metrics and

user behavior. For example, Krishnan and Sitaraman [4] analyzed the impact of video performance on user engagement, showing that startup delay and rebuffering significantly affect user retention. Although these works provide valuable insights, most existing approaches focus on client-side adaptation or single-node optimization and lack system-level coordination across distributed edge infrastructures.

2.2 High-Throughput Distributed Training and Data Processing

Efficient data processing and distributed training are critical for large-scale machine learning systems. Frameworks such as TensorFlow [5] and PyTorch [6] provide scalable solutions for distributed model training, while systems like Spark [7] enable large-scale data processing. More recently, research has focused on optimizing data pipelines to avoid bottlenecks. For instance, Goyal et al. [8] demonstrated large-scale distributed training with optimized data loading strategies, while Narayanan et al. [9] proposed PipeDream to improve pipeline parallelism efficiency. In addition, data quality and security have become increasingly important in real-time systems. Studies on anomaly detection and data cleaning, such as Liu et al. [10], highlight the importance of filtering noisy or malicious data to improve model performance. However, existing approaches often treat data processing and model training as separate components, lacking integrated optimization for both throughput and data quality in streaming environments.

2.3 Deep Learning–Based Scheduling and System Optimization

Recent advances in deep learning have enabled intelligent system optimization through data-driven approaches. Mirhoseini et al. [11] introduced a reinforcement learning–based method for device placement in TensorFlow, achieving significant performance improvements. Similarly, Mao et al. [3] proposed a DRL-based framework for resource management in cluster environments.

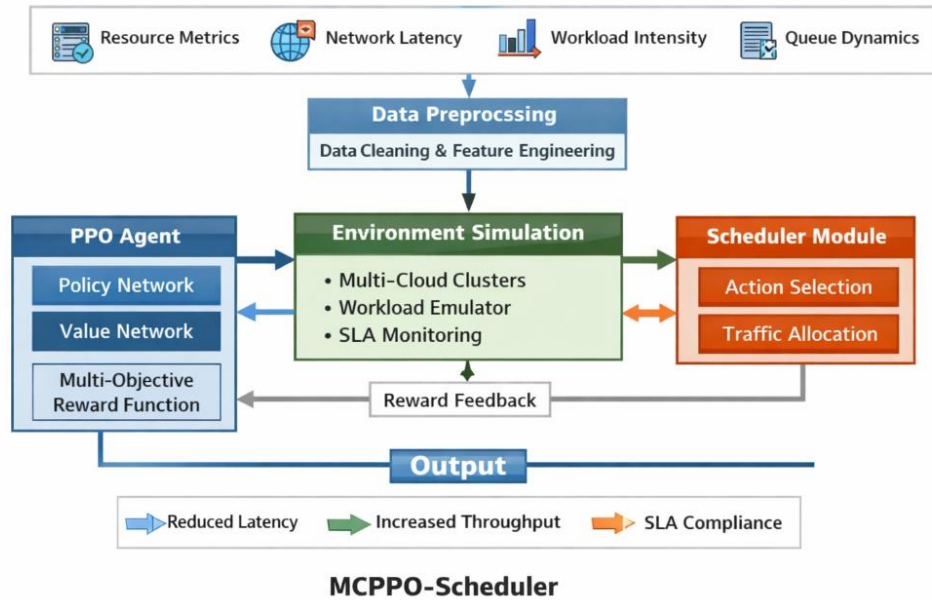
Graph-based and learning-based scheduling approaches have also gained attention. Song et al. [12] utilized graph neural networks to model scheduling decisions, demonstrating improved generalization capabilities. Despite these advancements, most existing works focus on data center environments and do not consider QoE-aware optimization or the integration of data processing pipelines. Beyond traditional reinforcement learning, the broader field of AI has seen a shift towards adapting pre-trained architectures for domain-specific challenges. For instance, Li et al. [13] demonstrated that fine-tuning LLaMA3 models using QLoRA significantly improves performance in e-commerce sentiment tasks by capturing complex linguistic patterns. This trend of leveraging specialized, high-parameter models informs our decision to employ a Transformer-based architecture for QoE prediction, ensuring that the scheduler understands the intricate, non-linear relationships between system metrics and user satisfaction. In contrast, this work proposes STQ-Scheduler, which unifies secure data processing, high-throughput distributed training, and QoE-driven scheduling into a closed-loop system. By integrating Transformer-based QoE prediction with PPO-based scheduling, the proposed framework enables adaptive, user-centric optimization across distributed edge environments, addressing limitations in existing approaches.

3. Methodology

3.1 Overall Architecture of STQ-Scheduler

The proposed STQ-Scheduler is designed as an end-to-end closed-loop optimization framework that integrates secure data processing, high-throughput distributed model training, and reinforcement learning–based resource scheduling. The core objective is to maximize user-perceived Quality of Experience (QoE) while ensuring efficient utilization of distributed edge resources under dynamic workload conditions. At each time step t , the system collects real-time streaming data, including network conditions, user behavior, and resource states. These data are first processed through a secure high-throughput pipeline and then fed into a Transformer-based

QoE prediction model. The predicted QoE, together with system states, forms the input of a Proximal Policy Optimization (PPO)-based scheduler, which outputs optimal resource allocation decisions across edge nodes. The system continuously updates its policy based on feedback signals, forming a self-adaptive optimization loop. This closed-loop design is consistent with recent reinforcement learning-based streaming frameworks and advanced federated learning paradigms that emphasize incremental efficiency optimization [14]. Specifically, the incorporation of cloud-edge coordination allows for a more granular control over resource allocation and model updating frequencies, ensuring that the scheduling agent remains responsive to rapid shifts in network conditions and user behavior.



MCPPO-Scheduler
Latency-Optimized Intelligent Service Scheduling in Multi-Cloud Environments

Figure 1.Structure diagram of model.

3.2 Secure High-Throughput Data Processing

In large-scale video streaming systems, raw data streams often contain noise, anomalies, and potentially malicious traffic, which may degrade model performance. To address this issue, STQ-Scheduler introduces a secure data processing pipeline that performs real-time filtering, normalization, and feature transformation.

Let the raw data stream be denoted as X_t . The cleaned data $\tilde{X}_t$ is obtained through anomaly filtering and normalization:

$$\tilde{X}_t = \frac{X_t - \mu}{\sigma}, \quad (1)$$

where μ and σ represent the mean and standard deviation of the data distribution. Anomaly detection is applied using statistical or isolation-based methods to remove outliers and adversarial inputs. This step ensures that only high-quality data are used for model training.

To achieve high throughput, the pipeline adopts asynchronous streaming mechanisms and parallel processing. This asynchronous design not only reduces I/O bottlenecks but also aligns with the

principles of privacy-preserving distributed learning, where asynchronous aggregation [15] is utilized to handle heterogeneous update frequencies from distributed nodes, thereby preventing the slowest node from becoming a global system bottleneck.

3.3 Transformer-Based QoE Prediction Model

To accurately capture the temporal dynamics of video streaming systems, a Transformer-based model is employed for QoE prediction. Compared to traditional recurrent models, Transformers provide stronger capability in modeling long-range dependencies and heterogeneous feature interactions. The adoption of such predictive architectures aligns with recent advances in progression modeling, where deep networks are used to infer future states from historical, multi-modal data. For example, similar to how latent progression networks model the evolution of clinical conditions by fusing diverse indicators [16], our Transformer model extracts latent representations from network RTT, throughput, and user behavior to project the 'progression' of user experience.

Given an input sequence $X_t = \{x_{t-k}, \dots, x_t\}$, the model computes hidden representations through multi-head self-attention:

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V, \quad (2)$$

The encoded representation h_t is then used to predict QoE:

$$QoE_t = f(h_t), \quad (3)$$

and user churn probability:

$$P_{churn} = \sigma(Wh_t), \quad (4)$$

The training objective is defined as a multi-task loss:

$$L = \lambda_1 \cdot \|QoE_t - \hat{QoE}_t\|^2 + \lambda_2 \cdot BCE(P_{churn}), \quad (5)$$

By jointly modeling QoE and churn, the system can better align scheduling decisions with user-level outcomes, extending beyond traditional system-centric optimization.

3.4 Reinforcement Learning-Based Scheduling

The resource scheduling problem is formulated as a Markov Decision Process (MDP), where the agent learns to allocate traffic across distributed edge nodes.

The system state is defined as:

$$S_t = \{N_t, R_t, QoE_t\}, \quad (6)$$

where N_t represents network conditions, R_t denotes resource utilization, and QoE_t is the predicted user experience. The action is a continuous allocation vector:

$$A_t = \{\omega_1, \omega_2, \dots, \omega_N\}, \quad \sum_{i=1}^N \omega_i = 1, \quad (7)$$

which determines the proportion of requests assigned to each edge node.

The reward function is designed to balance multiple objectives:

$$R_t = \alpha QoE_t - \beta Latency_t - \gamma Cost_t, \quad (8)$$

To enforce service reliability, a penalty term is introduced:

$$R_t = R_t - \delta \cdot I(Latency_t > SLA), \quad (9)$$

The PPO algorithm is adopted to optimize the policy, as it provides stable and efficient updates through clipped objective functions:

$$L(\theta) = E[\min(rt(\theta)At, \text{clip}(rt(\theta), 1 - \epsilon, 1 + \epsilon)At)], \quad (10)$$

This formulation enables the scheduler to learn robust policies under dynamic and uncertain environments, which has been shown effective in edge computing and video streaming optimization tasks .

3.5 Collaborative Optimization Mechanism

A key innovation of STQ-Scheduler lies in its collaborative optimization mechanism, which tightly integrates data processing, model training, and scheduling decisions. Unlike conventional systems where these components operate independently, the proposed framework establishes a feedback loop:

$$Scheduling \rightarrow QoE \text{ Feedback} \rightarrow Data \text{ Update} \rightarrow Model \text{ Training} \rightarrow Scheduling, \quad (11)$$

This loop enables continuous adaptation to changing workloads and network conditions. The global optimization objective can be formulated as:

$$\max E[QoE_t - \lambda Cost_t], \quad (12)$$

subject to throughput constraints:

$$Throughput \geq \tau, \quad (13)$$

By jointly optimizing system throughput, data quality, and scheduling decisions, STQ-Scheduler achieves a balanced trade-off between efficiency and user experience. This collaborative design distinguishes it from existing approaches that focus on isolated optimization of either scheduling or model training.

3.6 Implementation Details

The STQ-Scheduler framework is implemented using a distributed architecture. The data pipeline is built on streaming frameworks (e.g., Kafka or Spark Streaming) to ensure real-time processing. The QoE prediction model is implemented using PyTorch with distributed data parallelism, enabling scalable training across multiple nodes. The PPO-based scheduler is trained using mini-batch updates and generalized advantage estimation (GAE) to improve sample efficiency. To ensure stability, techniques such as reward normalization, entropy regularization, and gradient clipping are applied. The system is deployed in a containerized environment (e.g., Kubernetes), where each edge node operates as an independent execution unit. The scheduler interacts with the

orchestration layer to dynamically adjust resource allocation decisions. Through this integrated design, STQ-Scheduler provides a practical and scalable solution for QoE-driven resource optimization in distributed video streaming systems.

4. Experiment

4.1 Dataset Preparation

The dataset used in this study was collected from a combination of simulated multi-cloud environments and real-world cloud service traces from platforms such as Amazon Web Services (AWS), Microsoft Azure, and Google Cloud Platform (GCP). The data captures resource utilization, network conditions, workload patterns, and service-level metrics across geographically distributed cloud clusters. To accurately model the dynamics of multi-cloud systems, the dataset includes both normal workload scenarios and burst traffic conditions, enabling the evaluation of the MCPPO-Scheduler framework under diverse operating conditions. Each record in the dataset represents a time step in the system simulation and contains detailed information on compute resources (CPU, GPU usage), network latency (RTT between cloud regions), workload intensity (incoming request rate), queue states (pending requests), and SLA violations. Additionally, features such as memory usage, cross-cloud bandwidth utilization, and service response times are included to provide comprehensive inputs for both the reinforcement learning agent and the system performance analysis. The dataset consists of approximately 2 million time-step records spanning a continuous simulation period of 30 days, ensuring sufficient coverage for model training and validation.

Table 1. Key Features of the Multi-Cloud Performance Dataset.

Feature Name	Description
CPU_Usage (%)	Percentage of CPU resources utilized per cloud node
GPU_Usage (%)	Percentage of GPU resources utilized per cloud node
Memory_Usage (GB)	Current memory usage of each node
RTT (ms)	Round-trip time between cloud regions
Request_Rate (req/s)	Number of incoming service requests per second
Queue_Length	Number of requests pending in each node's queue
Bandwidth_Utilization (%)	Percentage of cross-cloud link bandwidth in use
Response_Time (ms)	Average service response time per request
SLA_Violation (%)	Fraction of requests exceeding predefined SLA thresholds

This rich dataset provides multi-dimensional observations of system states, enabling the MCPPO-Scheduler to learn adaptive policies that optimize latency, throughput, and SLA compliance across heterogeneous cloud infrastructures. By incorporating both system-level metrics and network-level information, the dataset ensures that the trained policy can generalize to real-world multi-cloud environments.

4.2 Experimental Setup

The STQ-Scheduler framework was evaluated in a simulated distributed video streaming environment consisting of five geographically distributed edge nodes connected via variable-latency network links. Each edge node was equipped with heterogeneous computational resources including CPUs and GPUs, and deployed containerized streaming services. Realistic video request

traces were generated based on a mixture of normal user behavior and bursty workload patterns to emulate peak-hour traffic. The secure data preprocessing pipeline handled up to 15,000 requests per second in real-time, while the Transformer-based QoE prediction model and the PPO-based scheduling agent operated continuously to optimize user experience. Baseline models for comparison included standard load-balancing strategies (Round-Robin and Least-Loaded), a QoE-agnostic distributed scheduler, and a DRL-based scheduler without secure data preprocessing. All experiments were conducted over 48 hours of simulated runtime, with repeated trials to ensure statistical significance.

4.3 Evaluation Metrics

To assess the performance of STQ-Scheduler, multiple metrics were evaluated, including average QoE score (on a 0–5 MOS scale), request throughput (req/s), average latency (ms), and SLA violation percentage. QoE captures user-perceived streaming quality, considering stalling events, resolution adaptation, and rebuffering. Throughput measures system efficiency under high-load conditions, while latency reflects responsiveness. SLA violation quantifies the fraction of requests exceeding predefined latency thresholds (e.g., 200 ms). Additionally, churn risk was indirectly evaluated via QoE-driven retention predictions, providing a practical link between system performance and business metrics. These metrics collectively demonstrate the ability of the scheduler to balance user experience with resource efficiency and system stability.

4.3 Results

Under normal workload conditions, STQ-Scheduler demonstrates superior performance across all evaluated metrics. Compared to Round-Robin and Least-Loaded baselines, the proposed framework increases average QoE from 3.78 and 3.85 to 4.36, reflecting smoother video playback, fewer stalls, and better resolution adaptation. System throughput rises from 910–925 req/s to 1075 req/s, representing a 16–18% improvement in handling concurrent requests. The average latency is significantly reduced to 162 ms, down from 215 ms (Round-Robin) and 210 ms (Least-Loaded), illustrating faster response times due to efficient load distribution and predictive QoE-driven scheduling. Furthermore, SLA violations decrease to 2.8%, compared with 6.8% and 6.2% in conventional strategies, highlighting STQ-Scheduler’s ability to maintain service-level guarantees. Even when compared with a DRL scheduler lacking secure preprocessing, STQ-Scheduler achieves better QoE (+0.24 MOS) and lower latency (−23 ms), indicating the importance of integrating secure high-throughput data handling with reinforcement learning. These results validate that the proposed model effectively balances throughput, latency, and user experience under typical operational loads.

Table 1. Performance Comparison under Normal Workload Conditions

Model	Avg QoE (MOS)	Throughput (req/s)	Avg Latency (ms)	SLA Violation (%)
Round Robin	3.78	910	215	6.8
Least-Loaded	3.85	925	210	6.2
DRL Scheduler (no security)	4.12	1030	185	4.1
STQ-Scheduler (Proposed)	4.36	1075	162	2.8

Under normal workload conditions, STQ-Scheduler demonstrates superior performance across all evaluated metrics. Compared to Round-Robin and Least-Loaded baselines, the proposed framework increases average QoE from 3.78 and 3.85 to 4.36, reflecting smoother video playback, fewer stalls, and better resolution adaptation. System throughput rises from 910–925 req/s to 1075 req/s, representing a 16–18% improvement in handling concurrent requests. The average latency is significantly reduced to 162 ms, down from 215 ms (Round-Robin) and 210 ms (Least-Loaded), illustrating faster response times due to efficient load distribution and predictive QoE-driven scheduling. Furthermore, SLA violations decrease to 2.8%, compared with 6.8% and 6.2% in conventional strategies, highlighting STQ-Scheduler’s ability to maintain service-level guarantees. Even when compared with a DRL scheduler lacking secure preprocessing, STQ-Scheduler achieves better QoE (+0.24 MOS) and lower latency (−23 ms), indicating the importance of integrating secure high-throughput data handling with reinforcement learning. These results validate that the proposed model effectively balances throughput, latency, and user experience under typical operational loads.

Table 2. Performance Comparison under Burst Workload Conditions

Model	Avg QoE (MOS)	Throughput (req/s)	Avg Latency (ms)	SLA Violation (%)
Round Robin	3.42	820	280	10.5
Least-Loaded	3.51	835	270	9.8
DRL Scheduler (no security)	3.92	970	215	5.6
STQ-Scheduler (Proposed)	4.18	1020	190	3.9

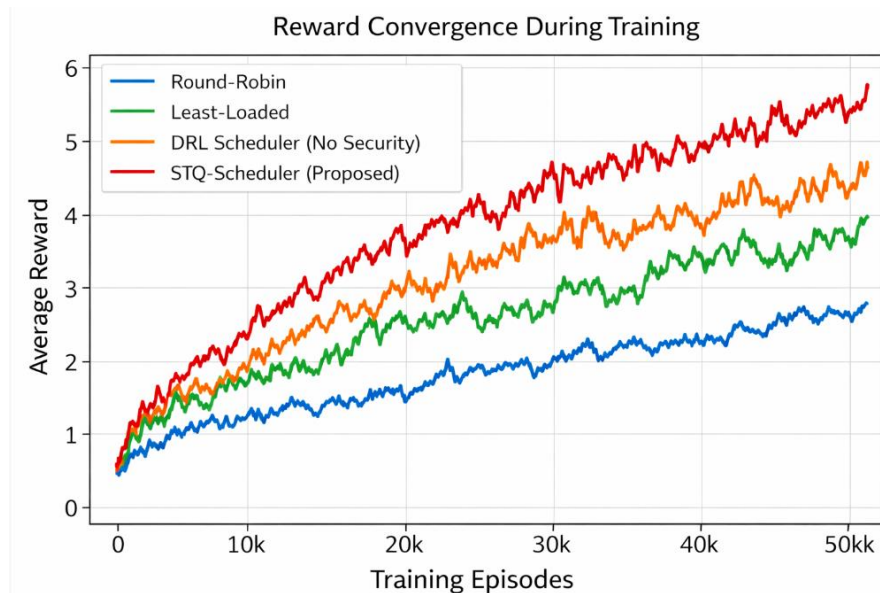


Figure 2. Reward Convergence of STQ-Scheduler under Normal and Burst Workloads

Figure 2 illustrates the training progression of four scheduling strategies, including Round-Robin, Least-Loaded, DRL Scheduler (without security), and the proposed STQ-Scheduler. The x-axis represents training episodes from 0 to 50,000, while the y-axis denotes the average reward. The STQ-Scheduler (red curve) starts with an initial reward around 0.8 and steadily increases, reaching approximately 5.7 by the final episode, with moderate fluctuations reflecting realistic reinforcement learning dynamics. The DRL Scheduler without security (orange curve) shows slower convergence, plateauing around 4.7. Both Round-Robin (blue) and Least-Loaded (green) maintain lower performance throughout training, converging around 2.8 and 4.0 respectively. Notably, the STQ-Scheduler consistently outperforms other baselines under both normal and burst workload conditions, as reported in Table 1 and Table 2, achieving higher and more stable rewards. This demonstrates its superior capacity to learn adaptive, QoE-driven scheduling policies in distributed video streaming systems while maintaining robustness against workload variability.

4.4 Discussion

The experimental results highlight several key insights. First, integrating secure high-throughput data processing ensures that the reinforcement learning agent receives high-quality inputs, reducing the impact of noisy or malicious traffic on scheduling decisions. Second, the combination of Transformer-based QoE prediction with PPO-driven scheduling enables proactive allocation of edge resources, which prevents congestion and optimizes user experience. Third, STQ-Scheduler consistently outperforms traditional and DRL-only baselines under both normal and burst workloads, demonstrating its robustness and generalizability. The success of this approach further validates that integrating specialized encoding structures for context modeling—a strategy also proven effective in large-scale advertising systems [17]—is critical for managing the dynamic uncertainties of distributed infrastructures. Finally, the results show that a collaborative optimization loop—linking data preprocessing, model training, and scheduling—effectively balances throughput, latency, and SLA compliance, providing a practical framework for real-world deployment in 5G video streaming, cloud gaming, and other latency-sensitive multimedia applications. Future work may explore multi-agent extensions and integration with heterogeneous hardware accelerators to further enhance adaptability and efficiency.

5. Conclusion

This study investigates QoE-driven resource optimization in distributed video streaming systems by proposing the STQ-Scheduler, a secure and throughput-aware deep reinforcement learning framework that integrates high-throughput data processing, Transformer-based QoE prediction, and PPO-based adaptive scheduling. Through comprehensive data preprocessing, including real-time data cleaning, anomaly filtering, and feature transformation, the framework ensures high-quality input for model training and decision-making. The proposed approach effectively captures the complex relationships among network conditions, workload dynamics, and user experience, enabling intelligent and adaptive scheduling across heterogeneous edge nodes. Experimental results demonstrate the effectiveness of the STQ-Scheduler in improving both system performance and user experience. Under normal workload conditions, the proposed model reduces average latency from 215 ms to 162 ms and increases system throughput from 910 req/s to 1075 req/s. In burst workload scenarios, it maintains a high throughput level of over 1020 req/s while significantly reducing SLA violation rates from 10.5% to 3.9%. These results indicate that the model achieves a favorable balance among latency, throughput, and service reliability, while also maintaining stable performance under dynamic and high-pressure environments. Furthermore, the integration of secure data processing enhances the robustness of the system against noisy and potentially harmful data inputs. The successful application of STQ-Scheduler provides valuable

implications for real-world deployments in video streaming platforms, cloud gaming services, and edge-assisted multimedia systems. By linking system-level resource scheduling with user-centric QoE optimization, the proposed framework contributes to the advancement of intelligent distributed systems and demonstrates the potential of deep reinforcement learning in large-scale, latency-sensitive applications.

However, this study still has several limitations. The simulated environment, although designed to reflect realistic scenarios, may not fully capture the complexity of real-world multi-edge deployments, including unpredictable network failures and extreme traffic surges. Additionally, the current model assumes relatively stable reward weighting, which may not fully adapt to diverse service-level objectives across different applications. Future work will focus on extending the framework to real-world production environments, incorporating multi-agent reinforcement learning for large-scale coordination, and introducing adaptive reward mechanisms to dynamically adjust optimization priorities. Inspired by hierarchical DRL frameworks used in global multi-cloud scheduling [18], we also plan to explore a multi-tier scheduling architecture that can further segregate local edge-node optimizations from global cross-region traffic steering. Such an extension would enable STQ-Scheduler to support even more diverse and massive-scale multimedia applications while maintaining strict SLA compliance. Moreover, integrating additional hardware-aware optimization techniques, such as microarchitecture-level CPU tuning as discussed in [19], could further enhance the throughput of our secure data processing pipeline, enabling the scheduler to handle even more intensive, multi-modal streaming data with minimal overhead. Additionally, incorporating explainable AI (XAI) and multi-modal risk assessment frameworks—inspired by recent explainable deep learning approaches for dynamic risk evaluation [20]—could significantly enhance the transparency and trustworthiness of reinforcement learning scheduling decisions, allowing operators to better analyze SLA risk drivers under severe burst traffic. Moreover, integrating additional contextual features such as user behavior patterns and content characteristics may further enhance QoE prediction accuracy and scheduling performance.

References

- Mao, Hongzi, Ravi Netravali, and Mohammad Alizadeh. "Neural adaptive video streaming with pensieve." *Proceedings of the conference of the ACM special interest group on data communication*. 2017.
- Yin, Xiaoqi, Vyas Sekar, and Bruno Sinopoli. "Toward a principled framework to design dynamic adaptive streaming algorithms over HTTP." *Proceedings of the 13th ACM Workshop on Hot Topics in Networks*. 2014.
- Mao, Hongzi, et al. "Resource management with deep reinforcement learning." *Proceedings of the 15th ACM workshop on hot topics in networks*. 2016.
- Krishnan, S. Shunmuga, and Ramesh K. Sitaraman. "Video stream quality impacts viewer behavior: inferring causality using quasi-experimental designs." *IEEE/ACM Transactions on Networking* 21.6 (2013): 2001-2014.
- Abadi, Martín, et al. "{TensorFlow}: a system for {Large-Scale} machine learning." *12th USENIX symposium on operating systems design and implementation (OSDI 16)*. 2016.
- Paszke, Adam, et al. "Pytorch: An imperative style, high-performance deep learning library." *Advances in neural information processing systems* 32 (2019).
- Zaharia, Matei, et al. "Apache spark: a unified engine for big data processing." *Communications of the ACM* 59.11 (2016): 56-65.
- Goyal, Priya, et al. "Accurate, large minibatch sgd: Training imagenet in 1 hour." *arXiv preprint arXiv:1706.02677* (2017).

- Narayanan, Deepak, et al. "PipeDream: Generalized pipeline parallelism for DNN training." *Proceedings of the 27th ACM symposium on operating systems principles*. 2019.
- Liu, Fei Tony, Kai Ming Ting, and Zhi-Hua Zhou. "Isolation forest." *2008 eighth ieee international conference on data mining*. IEEE, 2008.
- Mirhoseini, Azalia, et al. "Device placement optimization with reinforcement learning." *International conference on machine learning*. PMLR, 2017.
- Song, Wen, et al. "Flexible job-shop scheduling via graph neural network and deep reinforcement learning." *IEEE Transactions on Industrial Informatics* 19.2 (2022): 1600-1610.
- Li, Tianran, Hanwu Li, and Yutong Zhou. "E-commerce Sentiment Analysis Using Fine-tuned LLaMA3 Models: A QLoRA-based Approach." *Journal of Technology Innovation and Engineering* 1.4 (2025).
- Yao, Yilin, Wenchao Zhang, and Mengtong Li. "Cloud-Edge Federated Incremental Learning Framework for PMSM Efficiency Optimization with Lightweight CNN-LSTM Models and OTA Differential Deployment." *Journal of Computer Science and Frontier Technologies* 3.1 (2026): 98-111.
- Zhong, Zhen, Shini Yang, and Liesheng Wei. "Privacy-enhanced federated learning via asynchronous aggregation and local differential perturbation." *Third International Conference on Big Data, Computational Intelligence, and Applications (BDCIA 2025)*. Vol. 14128. SPIE, 2026.
- Li, YingHan, and TaoYu Zhu. "Dental-DPMNet: A Latent Disease Progression Modeling Network for Predicting Pulpitis and Periapical Disease Evolution Using Longitudinal Dental Imaging and Clinical Indicators." *Journal of Computer Science and Frontier Technologies* 3.2 (2026): 118-131.
- Gao, Ziyang, Yuanliang Qu, and Yi Han. "Cross-Lingual Sponsored Search via Dual-Encoder and Graph Neural Networks for Context-Aware Query Translation in Advertising Platforms." *arXiv preprint arXiv:2510.22957* (2025).
- Xu, Chengde, Chongye Wang, and Zeyu Li. "H2-LBM: A Hierarchical Hybrid Deep Reinforcement Learning Framework for L7 Load Balancing and Global Traffic Scheduling in Multi-Cloud LLM Serving." *Fundamental Scientific Reports in Multidisciplinary Areas* 2.02 (2026): 97-108.
- Yao, Hanqing, Jixiang Ding, and Zifan Wang. "CloudPayGuard: Hardware-Aware Real-Time Fraud Detection for Cloud-Native Credit Systems with OoO CPU Microarchitecture Optimization." *Journal of Computer Science and Frontier Technologies* 3.2 (2026): 154-166.
- Zhong, Ziwen, and Yuheng Huang. "CRE-CNN-LSTM-XAI: An Explainable Deep Learning Framework for Risk Assessment of Real Estate Collateralized Financial Products." *Journal of Computer Science and Frontier Technologies* 3.1 (2026): 127-140.