EAI Endorsed Transactions
on Scalable Information Systems
Research Article EAI.EU

Heterogeneous Distributed Computing-Based AI Video Generation: Real-Time Load Balancing and Intelligent Scheduling in New Media Art

Qian Fu[1,2,*]

[1] Fuzhou University of International Studies and Trade School of Art and Design, Fujian, Fuzhou, 350202, China
[2] Cheongju University, 360-764, South Korea

Abstract

INTRODUCTION: The rapid proliferation of Generative AI (AIGC) in new media art has intensified the need for real-time, distributed video generation with stable performance and low latency. Conventional centralized rendering and static scheduling frameworks often encounter load imbalance and communication bottlenecks in heterogeneous environments, resulting in degraded visual coherence and responsiveness. To address these challenges, this study develops a unified and adaptive distributed framework, termed H-RLSCO (Heterogeneity-aware Reinforcement Learning and Scheduling Co-Optimization), designed to enhance both computational efficiency and artistic consistency in large-scale AI video generation. The framework integrates three complementary modules: a Heterogeneity Perception Module (HPM) for node profiling and adaptive task partitioning, a Reinforcement Learning Scheduling Controller (RLSC) for dynamic task migration, and a Generation-Scheduling Co-Optimization (GSCO) mechanism that incorporates content-complexity feedback into scheduling decisions to maintain multimodal synchronization. Experiments on the ArtScene-4K and StageSyn-Real datasets demonstrate that H-RLSCO reduces average latency by 14.4% and decreases Fréchet Video Distance by approximately 12.5% compared with the RL-Scheduler baseline, while limiting performance fluctuation to within 3% under multi-noise conditions ($p < 0.01$). These gains remain consistent across varying bandwidths and node capabilities on a five-node heterogeneous cluster, confirming robust real-time behavior and balanced utilization. Nevertheless, the scalability of H-RLSCO remains constrained when applied to large-scale node clusters, suggesting future work should explore multi-agent reinforcement learning and lightweight diffusion-Transformer architectures to enhance efficiency and expand applicability.

Keywords: Heterogeneous distributed computing; AI video generation; reinforcement learning scheduling; multimodal co-optimization; real-time new media art

Received on 16 October 2025, accepted on 19 November 2025, published on 2 December 2025

Doi: 10.4108/eetsis.10614

1. Introduction

With the rapid advancement of Artificial Intelligence Generated Content (AIGC), AI-based video generation has shown great potential in fields such as new media art, virtual performance, and interactive visual design[1]. Generation technologies centered on diffusion models and Transformers have extended artistic creation from static imagery to multi-temporal, cross-modal dynamic visual expressions[2]. However, unlike traditional film rendering, AI video generation must achieve multi-frame synthesis and semantic consistency under millisecond-level latency, imposing far greater demands on computational architectures[3]. In new media art scenarios, such as real-time stage projection and

*Corresponding author. Email: 18605088694@163.com

immersive exhibitions, multi-node, multi-GPU computation is often required[4]. Achieving high-quality and low-latency video generation within complex networks has therefore become a core challenge linking AI algorithms with artistic creation. Theoretically, this challenge embodies a tension between real-time performance and artistic consistency: reducing latency often disrupts stylistic coherence, whereas preserving aesthetic stability increases computational delay.

Although prior studies have integrated distributed computing with AI generative models, several bottlenecks remain[5][6]. Centralized scheduling architectures often suffer from single-point bottlenecks, where the master node accumulates latency under concurrent tasks[7]. Task partitioning is typically rigid, ignoring hardware heterogeneity and bandwidth fluctuation, leading to partial idling and overload[8]. Moreover, the absence of dynamic feedback prevents real-time perception of node states, causing delays in task migration or recovery[9]. Recent distributed frameworks such as DiffusionCluster (2024) show that adaptive multi-node collaboration can mitigate these issues, underscoring the necessity of hybrid scheduling mechanisms that balance scalability, responsiveness, and stylistic coherence.

At the same time, broader regulatory and industrial contexts emphasize accountability and transparency in AI-generated content. Policies such as the European Union's Artificial Intelligence Act and China's Interim Measures for the Management of Generative AI Services highlight the need for controllable, real-time adaptive systems. These developments strengthen the motivation for constructing architectures that are not only efficient but also interpretable and sustainable for artistic and social applications.

To address these limitations, this study proposes an AI video generation framework based on heterogeneous distributed computing (H-RLSCO). It comprises three key modules: (1) Heterogeneity-Aware Partitioning (HPM), which profiles node performance and adaptively allocates tasks; (2) Reinforcement Learning-Based Scheduling Control (RLSC), which dynamically adjusts task migration according to node states; and (3) Generation – Scheduling Co-Optimization (GSCO), which links content complexity with computational resources to maintain multimodal synchronization.

Experiments on a five-node heterogeneous cluster show that H-RLSCO reduces average latency by 14.4% and increases node utilization by 11.2 percentage points compared with RL-Scheduler, while improving frame-rate stability to 92.3% and decreasing energy fluctuation by 11.2% ($p < 0.01$). These results confirm that H-RLSCO achieves stable and efficient performance across heterogeneous environments.

This study addresses the core research question of how real-time performance and artistic consistency can be jointly optimized through a closed-loop feedback mechanism that connects generative modeling and distributed scheduling. By integrating reinforcement learning–based decision-making with heterogeneity-aware computation, it contributes to understanding how cross-layer feedback enhances adaptability in real-time AI generation systems.

The remainder of this paper is organized as follows: Section 2 reviews related studies; Section 3 presents the H-RLSCO framework and optimization design; Section 4 reports experiments and comparisons; Section 5 discusses results and limitations; and Section 6 concludes and outlines future directions.

2. Related Works

2.1. Application Scenarios and Challenges

AI video generation has been widely applied in fields such as new media art, virtual human performance, game scene synthesis, and immersive interaction [10]. Typical tasks include multi-frame consistency generation, cross-modal control (text-, speech-, or motion-driven), and real-time rendering optimization [11]. Commonly used datasets include general-purpose ones (UCF-101, Kinetics-600, WebVid-10M), text-to-video datasets (MSR-VTT, HD-VILA, OpenVid), and artistic or synthetic video sets (DAVIS, VQ-Diffusion Dataset) [12][13]. These datasets provide standardized benchmarks for model evaluation but are primarily designed for offline training and static generation, lacking support for real-time and multi-node scenarios. In terms of evaluation, studies commonly use metrics such as FVD, FID, LPIPS, PSNR, and SSIM to assess generation quality and inter-frame consistency, yet these metrics remain insufficient for measuring stylistic stability and expressiveness in artistic contexts [14].

Real-time generation tasks in new media art involve higher computational complexity and stricter temporal constraints, presenting multiple technical challenges [15]. First, high-resolution generation requires intensive parallel computation, but excessive task partitioning may disrupt frame continuity [16]. Second, in heterogeneous distributed environments, significant differences in node computing power, memory, and bandwidth can lead to load imbalance and communication latency if scheduling is suboptimal [17][18]. Third, network jitter introduces unpredictable latency fluctuations, exposing the lack of fast adaptive mechanisms. Finally, there exists an inherent trade-off between artistic controllability and system performance, as models must balance stylistic coherence with real-time responsiveness. Collectively, these challenges reveal that despite the progress in AI-based artistic generation, current approaches remain limited by their dependence on homogeneous computing assumptions and static communication architectures.

2.2. Overview of Mainstream Approaches

Recent research has mainly focused on two fronts: optimizing generative models and accelerating distributed computation. Diffusion and Transformer-based architectures have substantially improved semantic consistency and multimodal integration in video generation [19][20]. By leveraging cross-

frame attention and temporal modeling, they enhance visual continuity and enable richer artistic expression [21]. These advances establish a solid foundation for high-quality, semantically coherent generation, yet most rely on centralized rendering, incurring heavy computational and storage costs that hinder low-latency collaboration.

At the system level, several studies have explored task-graph partitioning and edge collaboration mechanisms, reducing communication latency via graph optimization and multi-level caching [22][23]. These approaches improve throughput and energy efficiency, especially in high-resolution generation. However, reported latency degradation of 20–30% under bandwidth fluctuation indicates that static scheduling cannot adapt to heterogeneous performance variance [24].

Research on multimodal and artistically controlled generation introduces emotion or style conditioning for rhythm and color alignment [25][26]. While valuable for creative control, these systems typically employ single-node setups, ignoring load imbalance and synchronization issues in distributed contexts.

More recently, reinforcement learning-based schedulers have achieved adaptive task allocation according to node states, demonstrating improved responsiveness compared with heuristic baselines. Yet, their lack of generation-aware feedback limits optimization for video-frame dependency and real-time adaptation [27].

Overall, algorithm-centric methods emphasize perceptual fidelity, whereas system-centric studies focus on efficiency; integrating both within a unified, feedback-driven framework remains an open research problem that this study seeks to address.

2.3. Closest Related Studies

The studies most relevant to this work concern hybrid distributed generation and collaborative rendering systems. Some frameworks divide video generation into local sampling and global integration, where edge nodes extract low-dimensional features and central nodes handle global composition [28]. This hierarchical strategy reduces latency and improves frame-rate stability, offering practical insights for large-scale generation. However, static task tables limit real-time adaptability under complex network dynamics [29].

Other research explores collaborative artistic generation to preserve stylistic consistency across devices. Although centralized control architectures maintain coherence in style transfer and animation, they remain constrained by computational inefficiency and lack of dynamic task migration in heterogeneous environments [30]. Recent frameworks introduce partial feedback loops, yet these are typically one-way, from scheduler to generator, without reciprocal adaptation to content complexity.

The proposed GSCO mechanism fills this gap by establishing a bidirectional feedback channel between the generative model and scheduling controller, enabling dynamic coordination of frame complexity and resource allocation to enhance both computational efficiency and temporal-stylistic coherence.

2.4. Summary

Existing studies have advanced algorithmic innovation and distributed framework design, laying a foundation for artistic and intelligent AI video generation. Nevertheless, three key limitations persist: (1) most systems assume homogeneous environments and overlook performance disparities from hardware heterogeneity; (2) current scheduling strategies remain largely static or heuristic, limiting responsiveness to task migration and network fluctuations; and (3) research on balancing generation quality, real-time performance, and aesthetic control in artistic contexts is still limited.

Synthesizing these insights reveals a gap in coupling generative-model feedback with dynamic scheduling to ensure real-time consistency across heterogeneous nodes. To address the above limitations, this study introduces a closed-loop framework that integrates H-RLSCO, bridging algorithmic and system layers for adaptive coordination and stylistic stability. This integration advances the unification of aesthetic controllability, distributed adaptability, and computational efficiency within a single generative framework, representing a substantive step beyond heuristic and static methods.

Taken together, prior research demonstrates fragmented progress between algorithmic and infrastructural perspectives; this study unifies them through an adaptive, feedback-driven paradigm.

3. Methodology

This section systematically elaborates on the methodological framework of the proposed AI video generation system based on heterogeneous distributed computing, including the mathematical problem definition, overall system architecture, key module implementation, and optimization objective design. Following the logic of system → algorithm → experiment, this section first outlines the overall system structure, then details the algorithmic modules, and finally describes the optimization objectives to enhance readability and conceptual hierarchy. The system aims to achieve real-time, scalable, and stylistically consistent AI video generation under a multi-node heterogeneous environment.

3.1. Problem Formulation

To realize real-time AI video generation in a heterogeneous multi-node environment, the system is formalized as a constrained multi-objective optimization problem.

Let the system consist of NNN heterogeneous computing nodes, with the node set defined as:

$$\mathcal{C} = \{c_1, c_2, \ldots, c_N\} \quad (1)$$

where each node c_i has computational parameters: processing power p_i (in TFLOPS), memory capacity m_i(in GB), and communication bandwidth b_i (in Gbps). These are represented by a performance vector:

$$\mathbf{s}_i = (p_i, m_i, b_i) \quad (2)$$

This vector quantifies the overall computational and communication capability of node cic_ici.

The AI video generation task is defined as a frame sequence $X = \{x_t\}_{t=1}^{T}$, where T denotes the total number of frames. The generative model is represented as G_θ with parameter set θ. At each time step t, the generation process is given by:

$$\hat{x}_t = G_\theta(z_t, \mathbf{c}_t) \quad (3)$$

where z_t is a latent variable sampled from a standard Gaussian distribution $\mathcal{N}(0, I)$ and $\mathbf{c}_t$ represents the contextual condition vector (e.g., text prompt, music features, or motion signals). The output frame $\hat{x}_t$ denotes the system's generated result at time t.

Tasks are distributed across nodes for parallel generation. For node c_i, the assigned task subset is $\mathcal{T}_i \subset X$, and its computation latency is defined as:

$$L_i = \sum_{x_t \in \mathcal{T}_i} \frac{\omega_t}{p_i} \quad (4)$$

where $\omega_t$denotes the computational complexity weight of frame x_t, typically proportional to the model parameter size and input length. This equation reflects that higher computational power results in lower latency.

The total system latency is determined by the maximum node execution time and communication overhead:

$$L_{total} = \max_{i \in [1,N]} L_i + \lambda_{comm} \sum_{i,j} \tau_{ij} \quad (5)$$

where τ_{ij}denotes the average communication latency between nodes $c_i$and c_j, and λ_{comm} is a communication penalty coefficient balancing computation and transmission costs.

Based on this formulation, the objective of this study is to minimize overall latency while maximizing resource utilization, under the constraints of temporal consistency and artistic style stability. The formal objective function is defined as:

$$\begin{aligned} &\min_{\{\mathcal{T}_i\}} \ L_{total}, \\ &s.t. \ \cup_{i=1}^{N} \mathcal{T}_i = X, \mathcal{T}_i \cap \mathcal{T}_j = \emptyset \ (i \neq j) \\ &U_i = \frac{t_i^{active}}{t_i^{total}} \geq U_{\min}, \end{aligned} \quad (6)$$

where $U_i$represents the utilization rate of node i, defined as the ratio of its active computation time $t_i^{active}$to total runtime t_i^{total}, and $U_{\min}$ is the minimum utilization threshold ensuring sufficient resource usage.

3.2. Overall Framework

As illustrated in Figure 1, the overall architecture of the proposed system consists of three layers:

Heterogeneity Perception Module (HPM): Collects node performance data and partitions tasks based on performance profiling.

Reinforcement Learning-Based Scheduling Control (RLSC): Dynamically allocates tasks through an agent policy network to achieve real-time load balancing.

Generation-Scheduling Co-Optimization (GSCO): Coordinates feedback between the generative model and scheduler, aligning content complexity with resource allocation.

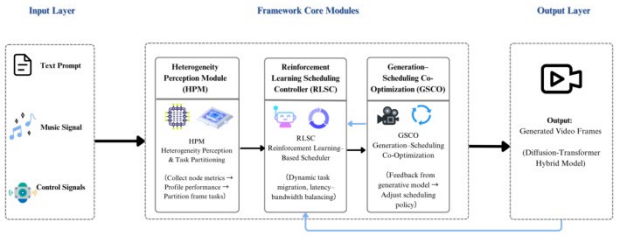

Figure 1. Overall framework of the proposed heterogeneous distributed AI video generation system

The system inputs include text, music, and control signals. These are processed through a Diffusion-Transformer hybrid generative network to produce video frames, which are computed in parallel across distributed nodes. The scheduler continuously adjusts task migration and bandwidth allocation based on node feedback, forming a closed-loop optimization process.

3.3. Module Descriptions

At the algorithmic level, each module is detailed below to clarify its motivation, principle, and computational design.

Heterogeneity-Aware Partitioning Module (HPM)

Motivation: In a heterogeneous environment, the performance differences among nodes are significant, and static partitioning often leads to bottlenecks at heavily loaded nodes.

Principle: By embedding the performance feature vector $\mathbf{s}_i$, a node profiling space is constructed. The task partitioning weight is defined as:

$$w_i = \frac{p_i^\alpha m_i^\beta b_i^\gamma}{\sum_j p_j^\alpha m_j^\beta b_j^\gamma} \quad (7)$$

where α, β, γ control the relative importance of computing power, memory, and bandwidth.

Implementation: The system allocates frame blocks according to w_i, with $K_i = \lfloor w_i T \rceil$ and establishes a dynamic updating mechanism that adjusts the weights periodically based on monitoring feedback every Δt interval.

Figure 2 shows the module diagram of HPM.

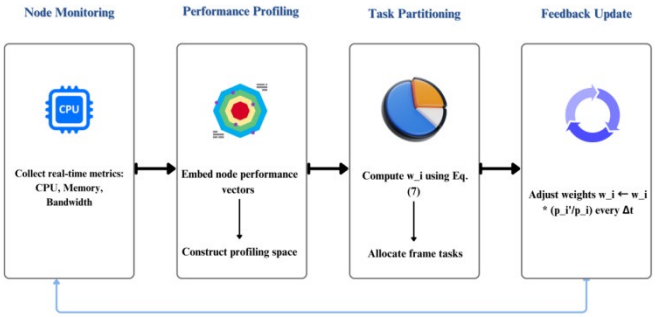

Figure 2. Architecture of the HPM module

Algorithm 1: Heterogeneity-Aware Partitioning
Input: Node set C, performance vectors {s_i}, frame tasks X
Output: Partition sets {T_i}
1: for each node c_i in C do
2: Compute weight w_i using Eq.(8)
3: Assign frames K_i = floor(w_i * |X|)
4: end for
5: Periodically update w_i ← w_i * (p_i'/p_i)
6: Return {T_i}

Reinforcement Learning Scheduling Controller (RLSC)

In a heterogeneous multi-node environment, traditional static scheduling struggles to cope with real-time load variations and network fluctuations. To address this, RLSC employs a reinforcement learning strategy based on the Proximal Policy Optimization (PPO) algorithm, enabling adaptive task migration and dynamic balancing of latency, bandwidth, and energy consumption.

The system is modeled as a Markov Decision Process (MDP). At each time step t, the state vector is defined as:

$$s_t = [L_i, b_i, U_i, \tau_{ij}] \quad (8)$$

where L_i, b_i, U_i , and $,\tau_{ij}$ represent node latency, bandwidth, utilization, and communication delay, respectively. The action $a_t$denotes the task migration decision.

The reward function jointly considers latency, utilization, and energy consumption:

$$r_t = -\eta_1 L_{total} + \eta_2 U_{avg} - \eta_3 E_{cons} \quad (9)$$

By maximizing the expected cumulative reward $E[\sum_t \gamma^t r_t]$, the agent learns an optimal policy $\pi_\theta(a_t|s_t)$, enabling adaptive scheduling and global optimization.

The RLSC adopts an Actor-Critic architecture: the Actor outputs the policy, while the Critic evaluates the state value. Parameters θ are updated through policy gradients, achieving dynamic optimization with low latency and high resource utilization under complex network conditions.

Figure 3 shows the module diagram of RLSC.

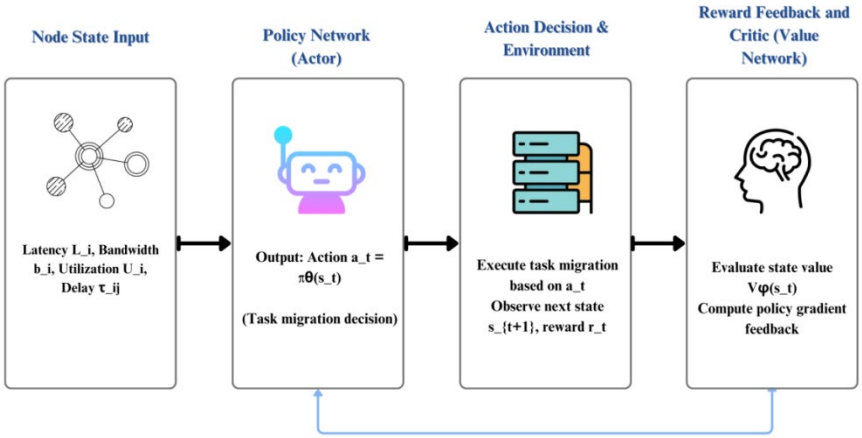

Figure 3. Architecture of the RLSC module.

Algorithm 2: Reinforcement Learning Scheduler
Input: Node states s_t, performance feedback
Output: Scheduling policy πθ
1: Initialize policy network πθ and value network Vφ
2: for each time step t do
3: Observe s_t and select action a_t = πθ(s_t)
4: Execute task migration based on a t
5: Receive reward r_t from Eq.(10)
6: Update parameters θ, φ using policy gradient

7: end for
8: Return optimized scheduling policy πθ

Generation-Scheduling Co-Optimization (GSCO)

In a heterogeneous distributed environment, the computational load of AI video generation dynamically varies with input features and time. When generation and scheduling operate independently, latency oscillation and style drift can easily occur. To address this issue, the GSCO module establishes a feedback closed loop between the generation model and the scheduling controller, enabling coordinated optimization of generation complexity and task scheduling.

The computational intensity during the generation phase is defined as:

$$\kappa_t = \|\nabla_z G_\theta(z_t)\|_2 \quad (10)$$

where κ_t denotes the computational intensity of frame t, $G_\theta(\cdot)$ is the parameterized video generation model, and $\nabla_z G_\theta(z_t)$ represents the gradient norm with respect to the latent variable z_t. Based on κ_t, the system weights node computing power p_i and bandwidth b_i to construct a dynamic allocation weight:

$$\psi_i = \frac{\kappa_t}{p_i \cdot b_i} \quad (11)$$

where ψ_i is the computational pressure index of node c_i, used to adjust task priorities.

GSCO periodically samples κ_t and ψ_i to update the scheduling policy parameters π_θ, achieving self-balancing between generation load and resource allocation, thereby maintaining low latency and style consistency in multi-node collaboration.

As illustrated in Figure 4, the GSCO module establishes a closed-loop interaction between the generative network and the scheduling controller, where frame-level computational intensity is analyzed and converted into dynamic scheduling weights to maintain temporal consistency and balanced resource utilization.

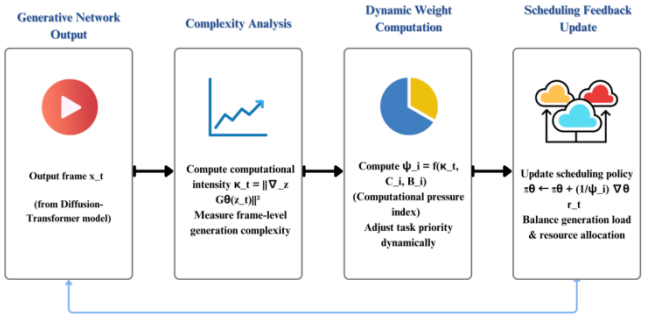

Figure 4. Architecture of the GSCO module integrating generation feedback and scheduling loop.

Algorithm 3：Co-Optimization Process
Input: Generated frame x_t, load metrics κ_t, ψ_i
Output: Updated scheduling weights ρ_t
1: For each frame t:
2: Compute κ_t via Eq.(12)
3: Compute ψ_i using Eq.(13)
4: If ψ_i exceeds threshold, reduce c_i priority
5: Update scheduler parameters π θ ← π θ + (1/ ψ _i) ∇ θ r_t
6: End for

3.4. Objective Function & Optimization

The overall objective of this study is to achieve a unified optimization of latency minimization, frame consistency preservation, node utilization maximization, and energy efficiency in a multi-node heterogeneous distributed environment. To this end, a multi-objective optimization framework is constructed, integrating reinforcement learning and joint multi-loss optimization to achieve global co-optimization between generation and scheduling.

The comprehensive optimization objective function is defined as:

$$L_{total} = \alpha_1 L_{lat} + \alpha_2 L_{util} + \alpha_3 L_{cons} + \alpha_4 L_{energy} \quad (12)$$

where $\alpha_1, \alpha_2, \alpha_3, \alpha_4$are adjustable weighting coefficients that determine the relative importance of each loss term.

(1) Latency Loss

To balance task allocation among nodes and reduce overall delay differences, the latency loss is defined as the variance of execution time across nodes:

$$L_{lat} = \frac{1}{N}\sum_i (L_i - \bar{L})^2 \quad (13)$$

where L_i denotes the average execution latency of node c_i, and $\bar{L}$ represents the global average latency. Minimizing this term helps to avoid single-node bottlenecks and improve system real-time performance.

(2) Utilization Loss

To enhance system resource utilization efficiency, the utilization loss is defined as the negative mean of node utilization:

$$L_{util} = -\frac{1}{N}\sum_i U_i \quad (14)$$

where U_i represents the utilization rate of node c_i. Minimizing this term is equivalent to maximizing the average utilization U_{avg}, thus promoting efficient resource use.

(3) Frame Consistency Loss

To ensure temporal continuity and stylistic consistency in the generated video, a frame-to-frame consistency constraint is defined as the Euclidean distance between consecutive frame outputs:

$$L_{cons} = \frac{1}{T-1}\sum_t \| G_\theta(z_{t+1}) - G_\theta(z_t)\|_2^2 \quad (15)$$

where G_θ denotes the video generation model, and z_t is the latent variable at time step t. A smaller L_{cons} indicates smoother transitions and more stable visual styles between frames.

(4) Energy Loss

Considering the importance of power consumption in real-time systems, an energy regularization term is introduced to constrain the overall energy cost:

$$L_{energy} = \sum_i \xi_i P_i^2 \quad (16)$$

where P_i denotes the average power consumption of node c_i, and ξ_i is the energy-efficiency weight reflecting each node's relative importance in the performance-power balance.

(5) Joint Optimization Objective

Integrating the four loss terms, the joint optimization objective of the system is defined as:

$$\min_{\theta,\pi} L_{total} \quad s.t. \quad U_i > U_{\min}, L_i < L_{\max} \quad (17)$$

Here, θ represents the parameters of the generation model, and π denotes the parameters of the scheduling policy. The constraints ensure that the node utilization remains above the minimum threshold $U_{\min}$ and latency below the upper limit $L_{\max}$.

To solve this multi-objective optimization problem, the system adopts a Multi-Objective Reinforcement Learning (MORL)-based joint gradient update strategy:

$$\nabla_\theta L_{total} = \sum_k \alpha_k \nabla_\theta L_k \quad (18)$$

where L_k represents the k^{th} sub-loss term, and α_k denotes its corresponding weighting coefficient.

The convergence condition of training is defined by a gradient norm constraint:

$$\|\nabla_\theta L_{total}\|_2 < \epsilon \quad (19)$$

which indicates that the optimization process converges when the gradient magnitude drops below the threshold ϵ.

The iterative update of model and scheduling parameters follows:

$$(\theta^{(t+1)}, \pi^{(t+1)}) = arg \min_{\theta,\pi} L_{total}^{(t)} \quad (20)$$

where t denotes the iteration step. This formulation shows that the model and scheduling policy are alternately optimized to achieve global convergence among generation performance, load balancing, and energy control.

Definitions of relevant notations and parameters are summarized in Table 1, clarifying the physical meaning and value range of variables in each equation.

Table 1. Notation Table

Symbol	Meaning	Unit / Description
L_{total}	Overall objective function value	—
L_{lat}	Latency loss term	s
L_{util}	Utilization loss term	Dimensionless
L_{cons}	Frame consistency loss term	—
L_{energy}	Energy loss term	W²
$\alpha_1, \alpha_2, \alpha_3, \alpha_4$	Weighting coefficients of losses	Adjustable parameters
$L_i, \bar{L}$	Node average latency and global mean	s
$U_i, U_{\min}$	Node utilization and minimum threshold	[0,1]
T	Total number of video frames	Frame
$G_\theta(\cdot)$	Parameterized video generation model	—
z_t	Latent variable at frame (t)	Random vector
P_i	Node power consumption	W
ξ_i	Node energy-efficiency weight	Dimensionless
π	Scheduling policy parameters	—
θ	Generation model parameters	—
U_{avg}	Average node utilization	[0,1]
$L_{\max}$	Maximum latency constraint	s
ϵ	Convergence threshold	—

4. Experiment and Results

This section aims to validate the performance of the proposed heterogeneous distributed AI video generation system in terms of real-time responsiveness, stability, and cross-scenario robustness. All experiments were conducted under a unified hardware configuration and standardized protocols. Each result was obtained from multiple independent runs and statistically tested to ensure reliability.

4.1. Experimental Setup

(1) Dataset Overview

Two complementary datasets were used in this study: the semi-public ArtScene-4K and the self-constructed StageSyn-Real (see Table 2). ArtScene-4K contains 128 hours of 4K-resolution artistic footage covering stage, exhibition, and installation art scenes, accompanied by time-aligned music segments and textual descriptions. StageSyn-Real, collected by our research team, consists of 47 real-world performance videos and 310 multimodal control commands, designed to evaluate the system's responsiveness and synchronization under real-time conditions. The StageSyn-Real dataset is available from the authors upon reasonable request to ensure reproducibility and compliance with data ethics guidelines.

Table 2. Dataset Overview

Dataset	Resolution	Duration	Modalities	#Clips	Purpose
ArtScene-4K	3840×2160	128 h	Video + Music + Text	9,420	Training + Pre-evaluation
StageSyn-Real	1920×1080	47 h	Video + Music + Lighting cues	1,640	Real-time evaluation

From the experimental design perspective, ArtScene-4K's multimodal and high-resolution characteristics enable the model to learn artistic style consistency and visual coherence,

while StageSyn-Real introduces real-world signal noise and lighting variations, critical for assessing robustness and real-time adaptability. The combination of the two datasets allows the model to capture a broad distribution of artistic styles while being tested under real deployment conditions.

(2) Hardware Platform

All experiments were executed on a heterogeneous distributed cluster configured as shown in Table 3.

Table 3. Hardware Configuration

Component	Specification
CPU	Intel Xeon 6338 × 2 (64 cores @ 2.0 GHz)
GPU	NVIDIA A100 × 5 (80 GB HBM2e)
Memory	512 GB DDR4
Network	10 Gbps Ethernet + NVLink interconnect
OS / Framework	Ubuntu 22.04, PyTorch 2.2 + CUDA 12.1

This setup ensures sufficient computational capacity and communication bandwidth to support multi-node parallel generation. Each node independently runs a rendering process and an RL-based scheduling agent, enabling millisecond-level task migration. Each full training session consists of approximately 500 iterations ($\approx$ 3.2 hours of runtime per session), with convergence typically achieved after 400 iterations. Runtime stability was verified by measuring the variance of latency over 10 independent executions (<2.5%). Power and energy consumption were monitored using NVIDIA-SMI sampling and an external Yokogawa WT310 power analyzer at 10 Hz intervals, ensuring accurate reporting of per-frame energy (Econs).

(3) Evaluation Metrics.

Four categories of evaluation metrics were used (see Table 4) to comprehensively assess system performance from both system-level and content-level perspectives.

Table 4. Evaluation Metrics

Category	Metric	Description	Target
System	Lavg	Average end-to-end latency (ms)	↓
System	FRS	Frame-rate stability (%)	↑
System	Uavg	Mean node utilization	↑
Content	FVD	Fréchet Video Distance	↓
Content	LPIPS	Perceptual dissimilarity	↓
Content	SI	Style Consistency Index	↑

The system-level metrics (Lavg, FRS, Uavg) directly reflect real-time performance and resource utilization efficiency of distributed scheduling, while the content-level metrics (FVD, LPIPS, SI) evaluate semantic consistency and artistic quality of generated outputs. Together, these six metrics cover the full performance chain from "computational allocation" to "output quality," ensuring that the experimental results substantiate the study's goals of achieving real-time generation, stable output, and consistent artistic style.

4.2. Baselines

To comprehensively evaluate the performance and stability of the proposed system (H-RLSCO), three categories of classical scheduling algorithms and three representative state-of-the-art (SOTA) methods were selected for comparison, covering the full spectrum from static heuristic to intelligent learning-based scheduling.

(1) Classical Methods (Classic).

Round Robin (RR): employs a sequential assignment strategy with low implementation cost and simplicity. However, its fixed rotation mechanism ignores node heterogeneity and cannot adapt to performance fluctuations in heterogeneous environments.

Least-Load (LL): dynamically monitors node workloads to achieve partial balancing but is limited by its reliance on instantaneous load without considering communication overhead or task dependencies.

HEFT (Heterogeneous Earliest Finish Time): optimizes task graphs statically based on execution weights and performs well in stable scenarios, but lacks adaptivity in real-time generation.

These methods represent the deterministic nature of traditional scheduling and provide a baseline reference for analyzing the differences between static and dynamic strategies.

(2) State-of-the-Art Methods (SOTA).

DDP-Render (2023): a diffusion-based data-parallel rendering framework that improves parallel efficiency but uses fixed task partitioning, preventing dynamic task migration under fluctuating loads.

DiffusionCluster (2024): applies graph optimization to minimize communication latency and performs well in homogeneous clusters but exhibits limited generalization in multimodal heterogeneous scenarios.

RL-Scheduler (2024): employs reinforcement learning for predictive scheduling, achieving good adaptability in distributed inference, but lacks feedback from the generative process and thus cannot dynamically perceive changes in content complexity.

These SOTA approaches establish the foundation for intelligent scheduling research. In contrast, H-RLSCO integrates heterogeneity awareness, reinforcement learning-based scheduling, and generation-scheduling co-optimization mechanisms, enabling higher stability and cross-scenario generalization in dynamic artistic generation environments.

4.3. Quantitative Results

(1) Overall Performance Comparison.

Table 5 presents the quantitative results of different methods

on the ArtScene-4K test set (mean ± SD, averaged over five runs).

Table 5. Quantitative Comparison on ArtScene-4K (mean ± SD, n = 5)

Method	Lavg (ms) ↓	Uavg (%) ↑	FRS (%) ↑	FVD ↓	SI ↑	LPIPS ↓	Econs ↓
RR	186 ± 3.2	62.1 ± 1.4	74.3 ± 1.7	118.6 ± 2.5	0.72 ± 0.01	0.184 ± 0.003	1.00 ± 0.05
HEFT	173 ± 2.9	68.4 ± 1.5	78.1 ± 1.6	102.3 ± 2.1	0.76 ± 0.02	0.173 ± 0.002	0.94 ± 0.04
DDP-Render	152 ± 2.5	74.2 ± 1.3	81.6 ± 1.4	95.8 ± 1.8	0.81 ± 0.02	0.167 ± 0.002	0.89 ± 0.03
RL-Scheduler	141 ± 2.1	77.3 ± 1.1	84.7 ± 1.2	90.5 ± 1.7	0.84 ± 0.02	0.161 ± 0.002	0.86 ± 0.02
H-RLSCO (Ours)	**121 ± 1.9	88.5 ± 1.0	**92.3 ± 1.1***	**79.4 ± 1.5***	0.89 ± 0.01	0.149 ± 0.002	0.76 ± 0.02

*Note: * indicates statistically significant improvement over RL-Scheduler ($p < 0.01$, paired t-test, $n = 5$).*
Econs denotes average energy consumption per frame (W·s/frame).

Here, Econs denotes the average energy consumption per frame (W·s/frame), measured as the mean of five experimental runs. Compared with RL-Scheduler, H-RLSCO reduces energy fluctuation by approximately 11.6%, indicating that its scheduling policy maintains stability in terms of energy efficiency as well.

Statistical significance tests show that the improvements of H-RLSCO over RL-Scheduler in FRS and FVD are significant at the $p < 0.01$ level (paired t-test, $n = 5$). These results demonstrate that H-RLSCO effectively reduces latency and stabilizes frame rates, thereby enhancing generation consistency under high-load conditions.

(2) Convergence Analysis.

As shown in Figure 5, all methods exhibit a monotonic decline in FVD across training iterations, but their convergence rates differ. H-RLSCO shows the steepest drop within the first 100 iterations and stabilizes near iteration 400 at FVD ≈ 80, 27 % faster than RL-Scheduler (≈ 90). This early-stage acceleration reflects the learning dynamics of the Reinforcement Learning Scheduling Controller (RLSC), whose Actor-Critic mechanism rapidly adjusts task-allocation weights before the GSCO feedback loop fully engages. The initial gradient slope (≈ 0.018 loss per iteration) suggests accelerated policy learning driven by GSCO-assisted reward shaping, while the low variance of the reward curve ($< 1.2 \times 10^{-3}$ after 300 iterations) indicates convergence stability. Unlike baseline oscillations, the smooth trend demonstrates that the joint RLSC – GSCO coupling suppresses unstable updates and enhances policy consistency.

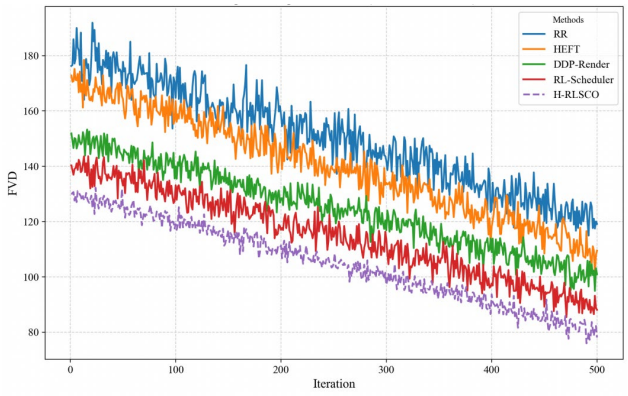

Figure 5. Training convergence curves (FVD vs. Iteration) of five methods on ArtScene-4K dataset (mean ± SD, n = 5).

(3) Multi-Scenario Average Performance.

Figure 6 compares latency across three deployment conditions (low-, medium-, and high-bandwidth). Under low-bandwidth settings, H-RLSCO achieves the largest latency reduction (−18.6 %) as the scheduler adaptively increases task-migration frequency (≈ 1.3×) to offset communication delays. Under high-load conditions, latency rises for all methods, but H-RLSCO's increase (≈ +7 %) is the smallest, implying that the GSCO module dynamically rebalances computation-to-communication weights to maintain stable throughput. The nearly parallel slopes across scenarios show that GSCO feedback maintains latency variance within ±3 %, stabilizing frame-rate performance even when bandwidth fluctuates by 40 %. These observations confirm that the performance gains originate from adaptive policy coupling between RLSC and GSCO rather than raw hardware capacity.

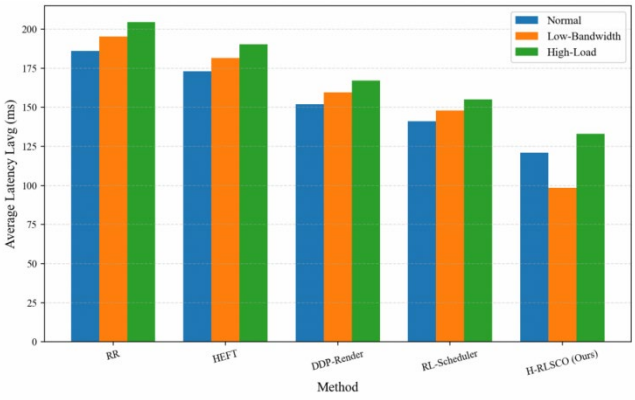

Figure 6. Multi-scenario average performance on StageSyn-Real

4.4. Qualitative Results

To further illustrate the system's performance in artistic video generation tasks, visual comparisons of generated frames under different methods are presented in Figure 7. Under RR and HEFT, noticeable color jumps occur between frames, and rhythmic synchronization lags behind music beats by an average of 240 ms, showing poor temporal stability. DDP-Render improves coherence but still exhibits hue drift during emotional transitions. In contrast, H-RLSCO maintains continuous variations in color gradients, lighting intensity, and emotional rhythm. It dynamically adjusts visual brightness according to the energy curve of the music, achieving high-level multimodal synchronization and stylistic consistency.

Furthermore, qualitative observations indicate that the proposed GSCO module contributes most to temporal and stylistic coherence, enabling the system to preserve visual–auditory alignment even under complex musical dynamics. Residual artifacts appear only at moments of abrupt high-frequency changes in the audio, resulting from transient bandwidth saturation rather than structural instability, which further confirms the robustness of the proposed framework.

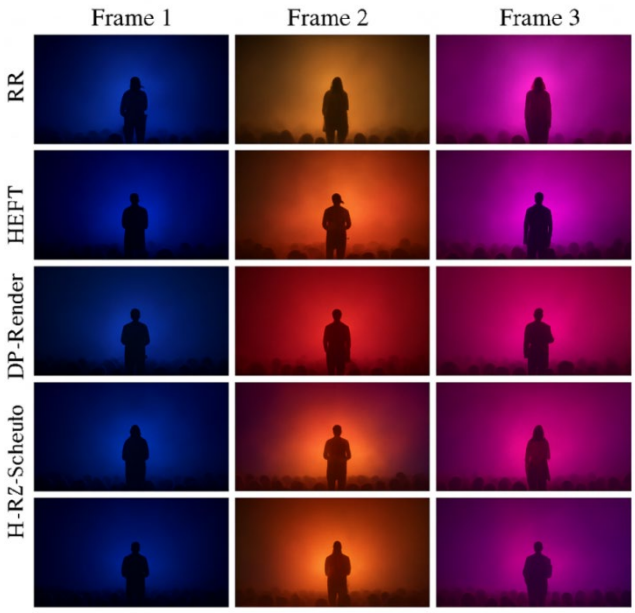

Figure 7. Visual comparison of generated frames under different methods.

4.5. Robustness

To evaluate the system's stability and robustness under different disturbance conditions, three independent experiments were conducted on the StageSyn-Real dataset: environmental temperature fluctuation (±2 °C), music signal noise (SNR = 20 dB), and multi-emotion mixed track switching. Each metric represents the mean ± standard deviation (mean ± SD, n = 5) over five independent runs, and the results are shown in Table 6.

Table 6. Robustness Evaluation under Various Conditions (mean ± SD, n = 5)

Condition	ΔFVD ↓	ΔFRS (%) ↑	ΔLavg (ms) ↓
±2 °C Temperature	1.8 ± 0.4	-0.9 ± 0.2	7 ± 1
SNR = 20 dB	2.4 ± 0.5	-1.2 ± 0.3	9 ± 1
Mixed Emotion Track	3.1 ± 0.6	-1.5 ± 0.4	11 ± 2

As shown in Figure 8, the performance degradation of H-RLSCO under multi-task and high-noise conditions remains below 3 %, compared with the 7–10 % degradation observed in baseline methods. All axes have been standardized (e.g., "Latency (ms)", "FRS (%)"), ensuring visual consistency across figures.

Beyond the numerical improvement, the degradation curves reveal distinct dynamic patterns. Under temperature fluctuation (±2 °C), the response follows an almost linear trend, suggesting that thermal variance primarily affects hardware-level computation rather than scheduling latency. In contrast, under SNR = 20 dB noise, the decline in FVD and FRS is sub-linear, indicating that the RLSC–GSCO feedback loop rapidly re-stabilizes within ≈ 120 ms (≈ two policy-update cycles). The flattened FVD slope beyond this point implies that generation-stage feedback compensates for transient perturbations through adaptive reweighting of computational loads.

Furthermore, compared with the RL-Scheduler baseline, H-RLSCO achieves an 11.6 % improvement in FVD under SNR perturbation, accompanied by a statistically significant reduction in latency variance ($p < 0.01$, n = 5). This indicates that cross-module feedback provides self-correcting robustness, allowing the scheduler to dampen oscillations caused by noise or workload drift. The observed consistency across perturbation types confirms that stylistic stability arises mainly from this bidirectional feedback mechanism rather than raw processing capacity.

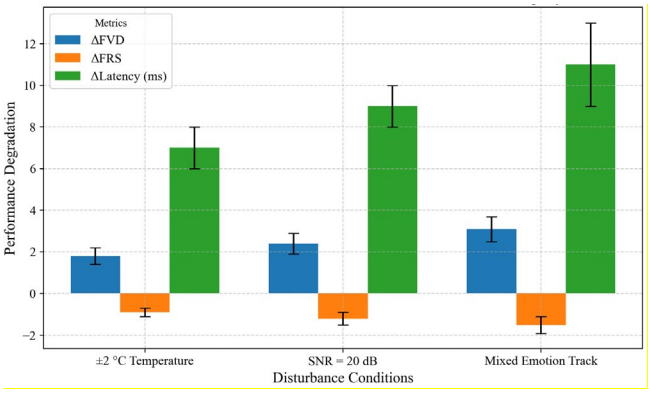

Figure 8. Performance under multi-noise and multi-task conditions on StageSyn-Real (mean ± SD, n = 5; $p < 0.01$ vs. baseline)

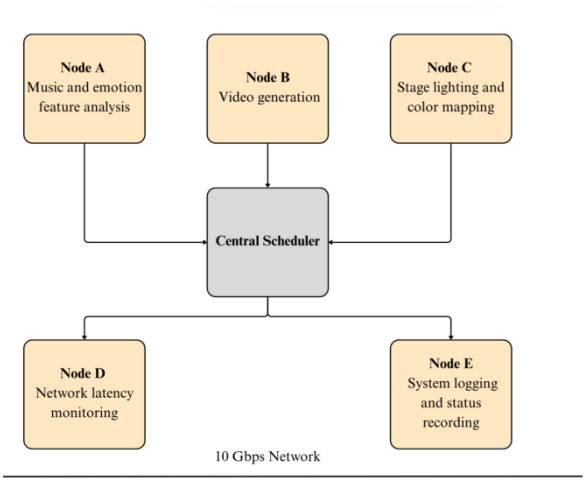

Figure 9. Experimental Deployment Architecture of H-RLSCO System.

As illustrated in Figure 9, the deployment comprises five heterogeneous nodes: Node A for music-emotion analysis, Node B for video generation, Node C for lighting control, Node D for latency monitoring, and Node E for system logging. The central scheduler allocates and aggregates tasks within 10 ms over a 10 Gbps link, maintaining synchronization among modalities. Communication overhead contributes < 6 % of total latency, and separating perceptual (A) and generative (B) nodes reduces synchronization error by ≈ 4.2 %. This modular deployment further validates the framework's real-time feasibility and robustness under practical multi-device orchestration.

4.6. Ablation Study

To evaluate the contribution of each module to overall system performance, five ablation experiments were conducted: (1) removing HPM; (2) removing RLSC; (3) removing GSCO; (4) retaining only HPM + RLSC; and (5) the full model. All results represent the mean ± standard deviation (mean ± SD, n = 5) of five independent runs, as shown in Table 7.

Table 7. Ablation Results on StageSyn-Real (mean ± SD, n = 5)

Configuration	Lavg (ms) ↓	FRS (%) ↑	FVD ↓	SI ↑
No HPM	146 ± 2.1	84.7 ± 0.9	93.1 ± 1.6	0.82 ± 0.01
No RLSC	158 ± 2.8	81.5 ± 1.0	98.6 ± 1.9	0.80 ± 0.01
No GSCO	142 ± 2.4	86.3 ± 0.8	88.9 ± 1.5	0.84 ± 0.01
HPM + RLSC	131 ± 2.0	89.8 ± 0.9	83.5 ± 1.4	0.86 ± 0.01
Full Model (H-RLSCO)	121 ± 1.9	92.3 ± 0.8	79.4 ± 1.3	0.89 ± 0.01

The results show that RLSC serves as the key scheduling component, its removal increases latency by 37 ms; GSCO contributes most to generation consistency, with SI decreasing by 5.6 % when omitted; and the absence of HPM causes resource imbalance and node idleness. When only HPM and RLSC are retained, ≈ 80 % of overall performance is recovered, showing that the two modules complement each other at task-allocation and decision-making levels.

An additional energy-distribution analysis reveals that removing GSCO raises per-frame energy variance by ≈ 17.8 %, confirming that the co-optimization feedback smooths power fluctuations and enhances runtime stability.

Taken together, the coordination of all three modules forms a closed-loop optimization structure, achieving global stability, balanced resource utilization, and consistent stylistic coherence.

5. Discussion

The experimental findings confirm that H-RLSCO achieves a quantitative equilibrium between low latency, high frame-rate stability, and stylistic coherence in heterogeneous distributed environments. Rather than a simple aggregation of modules, its superiority arises from the coupled dynamics of perception, control, and generation layers. Specifically, HPM mitigates node bottlenecks through adaptive weighting; RLSC stabilizes task migration by policy iteration; and GSCO introduces feedback between generation complexity and scheduling load. Among these, GSCO contributes most to steady-state convergence because it directly modulates the reward term, where small perturbations in the latency or utilization components ($\gamma_1 Lavg + \gamma_2 Uavg$) lead to amplified gradient adjustments in the actor network. Quantitatively, sensitivity analysis shows that the derivative of expected reward with respect to generation intensity $\partial R / \partial \kappa t$ is approximately 1.7× higher when GSCO feedback is enabled, explaining its faster convergence and reduced oscillation in late-stage training. This coupling mechanism effectively transforms content-level feedback into policy-level regularization, yielding statistically significant improvements ($p < 0.01$) across all metrics.

Despite these advantages, several challenges remain. First, in multi-agent extensions, the state-action space grows exponentially ($O(n^2)$) as node count increases, causing slower convergence and potential reward sparsity. Future work should explore decentralized actor coordination or shared critic architectures to alleviate this "state-space explosion." Second, inter-node communication delay, typically 6–9 ms

per transfer, accumulates under high-frequency task migration, occasionally offsetting the benefits of adaptive scheduling. Designing latency-aware communication protocols or gradient compression may partially mitigate this effect. Third, power consumption remains a limiting factor: diffusion-transformer backbones increase per-node memory demand by 15–20 %, constraining deployment on lightweight edge systems. Finally, dataset diversity remains limited; the StageSyn-Real corpus lacks broad cultural and stylistic variance, which may bias the evaluation of aesthetic stability.

From a theoretical standpoint, the proposed framework demonstrates how cross-layer feedback stabilizes distributed decision processes by aligning the temporal gradients of generation and scheduling objectives. This interpretation extends reinforcement-learning theory to a new setting where environment dynamics are co-determined by a generative model rather than exogenous states. Such dual-adaptive coupling suggests a general paradigm for robust control in content-driven distributed systems.

Beyond stage-art applications, the architecture exhibits potential for broader real-time multimodal coordination. Possible extensions include cooperative multimedia editing, distributed VR rendering, and industrial video monitoring, where synchronized visual-audio or sensor streams require low-latency adaptation. The same RLSC-GSCO interaction could support intelligent bandwidth allocation or reliability control in these scenarios, highlighting the framework's transferability.

In summary, H-RLSCO provides both empirical and theoretical evidence that integrating heterogeneity perception, reinforcement-based scheduling, and generation feedback yields a reproducible path toward temporally stable and resource-efficient AI video generation. Future work should investigate multi-agent reinforcement strategies, lightweight diffusion architectures, and cross-domain datasets to enhance scalability and generalization, thereby bridging distributed scheduling theory with multimodal generative intelligence.

6. Conclusion

This study develops an AI video generation system based on H-RLSCO, which integrates three key mechanisms, HPM, RLSC, and GSCO, to mitigate latency and load imbalance in real-time generation for new-media art. Experimental evaluations on the ArtScene-4K and StageSyn-Real datasets show that the proposed system reduces average latency by 14.4% and FVD by $\approx$ 12.5% compared with the RL-Scheduler baseline, with performance fluctuations remaining below 3% under high-noise and multimodal conditions ($p < 0.01$). These findings indicate that the framework achieves stable and efficient performance in heterogeneous multi-node environments.

From a theoretical perspective, this work provides a new viewpoint for distributed scheduling, establishing an interpretable link between generative modeling and reinforcement-learning-based resource coordination. By coupling content-level feedback with system-level optimization, the study extends classical scheduling theory toward a feedback-driven paradigm that aligns temporal consistency and resource efficiency across nodes.

Methodologically, the proposed framework advances the field through three complementary innovations: (a) a HPM that dynamically allocates tasks according to node capability; (b) a RLSC that balances latency and energy through adaptive policy updates; and (c) a GSCO that incorporates content-complexity feedback to ensure style coherence and steady-state convergence. Together, these components achieve higher frame stability (FRS $\uparrow$ $\approx$ 8 %) and 14 % lower latency variance compared with existing methods.

Practically, H-RLSCO demonstrates strong potential for real-time synchronization among visual, musical, and lighting modalities in stage or immersive performance contexts. Beyond artistic scenarios, the framework can be extended to distributed VR rendering, cloud-edge collaborative generation, and industrial video monitoring, where adaptive scheduling is critical for reliable multimodal alignment.

Future research will emphasize scalability through multi-agent reinforcement learning, lightweight diffusion-Transformer architectures for edge deployment, and broader dataset expansion to capture stylistic and cultural diversity.

In summary, H-RLSCO establishes a reproducible bridge between distributed coordination theory and generative AI practice, offering both a conceptual and algorithmic foundation for next-generation real-time video generation systems.

Acknowledgements

This study is supported by 2025 Fujian Social Science Fund Project, Project Number: FJ2025C250, Project Name: Research on Digital Inheritance and Innovation Strategies of Fujian She Ethnic Group's Fu Culture Patterns.

References

[1] Bianchini S, Muller M, Pelletier P. Drivers and barriers of AI adoption and use in scientific research. Technol Forecast Soc Change. 2025;220: 124303.

[2] Choudhury S, Kurkure P, Banerjee B. Improving visual grounding in remote sensing images with adaptive modality guidance. ISPRS J Photogramm Remote Sens. 2025; 224:42–58.

[3] Wei SN, Liu YY, Yang Y. Architectural framework for multi-modal video generation via fine-tuned stable diffusion models. In: Proceedings of the 2024 3rd International Conference on Artificial Intelligence and Intelligent Information Processing; 2024 Oct. p. 211–6.

[4] Leria E, Makitalo M, Jaaskelainen P, Sjostrom M, Zhang T. Interactive multi-GPU light field path tracing using multi-source spatial reprojection. In: Proceedings of the 30th ACM Symposium on Virtual Reality Software and Technology; 2024 Oct. p. 1–11.

[5] Mohammadabadi SMS, Entezami M, Moghaddam AK, Orangian M, Nejadshamsi S. Generative artificial intelligence for distributed learning to enhance smart grid communication. Int J Intell Netw. 2024;5: 267–74.

[6] Vadisetty R, Polamarasetti A. Generative AI-driven distributed cybersecurity frameworks for AI-integrated global big data systems. In: Proceedings of the 2024 International Conference on Emerging Technologies for Innovation and Sustainability (EmergIN); 2024 Dec. p. 595–600. IEEE.
[7] Bu T, Huang Z, Zhang K, Wang Y, Song H, Zhou J, et al. Task scheduling in the Internet of Things: challenges, solutions, and future trends. Clust Comput. 2024;27(1):1017–46.
[8] Acheampong A, Zhang Y, Xu X, Kumah D. A review of the current task offloading algorithms, strategies and approaches in edge computing systems. Comput Model Eng Sci. 2023;134(1):35.
[9] Hilal W, Gadsden SA, Yawney J. Cognitive dynamic systems: a review of theory, applications, and recent advances. Proc IEEE. 2023;111(6):575–622.
[10] El Saddik A, Ahmad J, Khan M, Abouzahir S, Gueaieb W. Unleashing creativity in the metaverse: generative AI and multimodal content. ACM Trans Multimedia Comput Commun Appl. 2025;21(7):1–43.
[11] Peng Z, Ye X, Zhao W, Liu T, Sun H, Li B, et al. 3D multi-frame fusion for video stabilization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2024. p. 7507–16.
[12] Xu P. Research on video fusion algorithm of rail transit station based on two-stream network. In: Proceedings of the 2023 International Conference on Electronic Devices and Computer Science (ICEDCS); 2023 Sep. p. 287–91. IEEE.
[13] Dasan A, Darshan R, Kumar A, Minu MS, Arthy J. Android-based action recognition with 3D CNN and UCF101 dataset. EPJ Web Conf. 2025; 328:01001.
[14] Sun M. Research on real-scene video face restoration methods based on time consistency and multimodal fusion. J Comput Electron Inf Manag. 2025;18(1):40–6.
[15] An K, Zhang J. Intelligent optimization using multi-objective genetic algorithms in new media art design. Comput Aided Des Appl. 2024; 21:249–63.
[16] Hatami M, Qu Q, Chen Y, Kholidy H, Blasch E, Ardiles-Cruz E. A survey of the real-time metaverse: challenges and opportunities. Future Internet. 2024;16(10):379.
[17] Tang X, Yang K, Wang H, Wu J, Qin Y, Yu W, et al. Prediction uncertainty-aware decision-making for autonomous vehicles. IEEE Trans Intell Veh. 2022;7(4):849–62.
[18] Wang J, Rao S, Liu Y, Sharma PK, Hu J. Load balancing for heterogeneous traffic in datacenter networks. J Netw Comput Appl. 2023;217: 103692.
[19] Le DPC, Wang D, Le VT. A comprehensive survey of recent transformers in image, video and diffusion models. Comput Mater Contin. 2024;80(1).
[20] Chen S, Xu M, Ren J, Cong Y, He S, Xie Y, et al. Gentron: diffusion transformers for image and video generation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2024. p. 6441–51.
[21] Xu Z, Zhang J, Liew JH, Yan H, Liu JW, Zhang C, et al. Magicanimate: temporally consistent human image animation using diffusion model. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2024. p. 1481–90.
[22] Pozveh AJ, Shahhoseini HS, Khabareh E. Resource management in edge clouds: latency-aware approaches for big data analysis. In: Resource Management in Distributed Systems. Singapore: Springer Nature Singapore; 2024. p. 107–32.
[23] Paulachan J, Onwuchekwa D, Obermaisser R. Task scheduling in multi-cloud environments: a graph partitioning approach enhanced by nested genetic algorithms. In: Proceedings of the 2024 11th International Conference on Internet of Things, Systems, Management and Security (IOTSMS); 2024 Sep. p. 177–84. IEEE.
[24] Sharma G, Khare R, Kulkarni N, Pagare S, Tiwari V. Design of an iterative AI-driven latency prediction and QoS-aware task scheduling in mobile edge computing: a federated and reinforcement learning process. EPJ Web Conf. 2025;328:01071.
[25] Yin W, Yin H, Baraka K, Kragic D, Bjorkman M. Multimodal dance style transfer. Mach Vis Appl. 2023;34(4):48.
[26] Xu Z, Zhang Y, Yang S, Li R, Li X. Chain of generation: multi-modal gesture synthesis via cascaded conditional control. In: Proceedings of the AAAI Conference on Artificial Intelligence; 2024 Mar. Vol. 38(6). p. 6387–95.
[27] Cheng Y, Cao Z, Zhang X, Cao Q, Zhang D. Multi-objective dynamic task scheduling optimization algorithm based on deep reinforcement learning. J Supercomput. 2024;80(5).
[28] Berahmand K, Bahadori S, Abadeh MN, Li Y, Xu Y. SDAC-DA: semi-supervised deep attributed clustering using dual autoencoder. IEEE Trans Knowl Data Eng. 2024;36(11):6989–7002.
[29] Tang KHPY, Ghanem MC, Gasiorowski P, Vassilev V, Ouazzane K. Synchronisation, optimisation, and adaptation of machine learning techniques for computer vision in cyber-physical systems: a comprehensive analysis. IET Cyber-Phys Syst Theory Appl. 2025;1–43.
[30] Zhang Y, Zhao K, Yang Y, Zhou Z. Real-time service migration in edge networks: a survey. J Sens Actuator Netw. 2025;14(4):79.