Spatiotemporal GAN-Based Real-Time Animation and Multi-Modal Interaction Optimization for Virtual Reality

Zheng Wang
Department of Digital Media Animation, Shandong Media Vocational College, Jinan 250000, China
E-mail: ZhenggWangg@outlook.com

Keywords: Virtual reality, generative adversarial networks, real-time animation composition, multimodal interaction

Received: September 4, 2025

Abstract: At present, animation generation and multi-modal interaction in virtual reality environments still face problems such as low generation quality, poor real-time performance and insufficient fusion between modes, which seriously restrict the authenticity and interaction efficiency of immersive experiences. A real-time synthesis and multi-modal interactive optimization method of generative adversarial network animation for a VR environment is proposed. In animation synthesis, a generation architecture with a spatiotemporal consistency adversarial training mechanism is constructed, and a multi-scale feature fusion strategy is combined to achieve high-quality and low-latency animation generation. Experiments were conducted on the VR (Virtual Reality)-Gesture-Voice dataset (60,000 training samples, 15,000 testing samples) and benchmarked against state-of-the-art (SOTA) models including VideoGAN and StyleGAN3. Key results: For the isolated GAN synthesis module: average rendering frame rate = 23.7 FPS (58% higher than VideoGAN), synthesis delay ≤ 4.5 ms; For the end-to-end VR system: average rendering frame rate = 85–92 FPS (meeting VR's ≥72 FPS standard), end-to-end latency ≤17 ms (51% lower than StyleGAN3). In the real-time synthesis test of generative adversarial network (GAN) animation in the VR environment, two sets of key metrics are reported to clarify different system scopes: For the isolated GAN animation synthesis module (excluding end-to-end transmission and rendering), the improved algorithm achieved an average rendering frame rate of 23.7 FPS (58% higher than the traditional method) and controlled the synthesis delay within 4.5 Ms. Regarding system resource usage, GPU (Graphics Processing Unit） memory consumption is reduced by 0.6 GB, model reasoning time is reduced by 49.5%, and 85% real-time rendering efficiency can still be maintained at 8K resolution.

Povzetek:

1 Introduction

Under the increasingly mature and wide application of virtual reality technology, immersive experience has become one of the important criteria for users to evaluate the advantages and disadvantages of VR systems [1, 2]. Virtual reality is a visual simulation technology and a comprehensive system that emphasizes real-time and interactivity [3]. Generative Adversarial Networks (GANs) have become foundational for dynamic animation synthesis. Early works like VideoGAN pioneered video sequence generation using 3D convolutional layers to capture spatiotemporal features, but it suffered from high latency (≥42 ms) and poor adaptability to VR's dynamic scenes—limitations that make it unsuitable for real-time interaction [4]. In key industries such as game entertainment, digital film and television, distance education, smart medical care, etc., the enabling effect of VR has become increasingly prominent. Users are no longer satisfied with the viewing or operation experience from a single perspective but hope to achieve a more natural, efficient and intelligent interactive process through multi-modal information interaction [5, 6]. The VR system still faces several technical problems that must be solved urgently. One of the core bottlenecks is the real-time generation and efficient interaction of animation content [7]. Can spatiotemporal consistency constraints integrate into GANs enhance VR animation realism (e.g., reduce jitter and dislocation) without sacrificing real-time performance (e.g., increasing latency)? Can a cross-modal attention alignment model improve the semantic consistency between multi-modal inputs (gesture, voice, vision) and thus enhance interaction accuracy and user satisfaction? [8, 9]

Our method addresses these gaps by reducing GPU memory consumption by 0.6 GB (vs. Instant NGP) while maintaining 85% real-time rendering efficiency at 8K resolution [10, 11]. GAN (Generative Adversarial Networks)-based video synthesis: GANs for video generation, such as VideoGAN and ST-GAN, address temporal consistency but lack optimization for VR's low-latency requirements [12, 13]. The real-time animation synthesis technology based on the generative adversarial network has become a key technical path to break through traditional animation's bottleneck and improve the real-time interaction ability [14]. The VideoGAN model from 2018, evaluated on the VR-Gesture-Voice dataset, achieved a Fréchet Inception Distance (FID) of 45.2, an average frame rate (FPS) of 48, a latency of 42 milliseconds, a temporal consistency score of 0.68, and a

semantic alignment accuracy of 78.5%. In 2021, StyleGAN3 improved these metrics on the same dataset, with an FID of 38.9, an average FPS of 52, a latency of 35 milliseconds, a temporal consistency score of 0.75, and a semantic alignment accuracy of 81.2% [15, 16]. Speech signals from sensors, human posture data collected by motion capture devices, user facial expressions and other information constitute a high-dimensional and heterogeneous data system. On this basis, the rapid fusion of data and semantic unification have become the key factors affecting the interaction accuracy and system response speed [17].

2 Research on optimization of generative adversarial network architecture in virtual reality environment

2.1 Construction of spatio-temporal consistency adversarial training framework

In virtual reality environment, real-time synthesis of animation is one of the core technologies to build immersive experience [18]. Temporal incoherence as shown in equations (1) and (2), At is the animation frame generated at time step t; Xt is input data at time step t; Ht−1 is the hidden state of the previous time step; Wg is the generated network weight; Et is the generation error. Lc is the generation reconstruction loss; D is the discriminator function; λ1 is the regularization coefficient; R is the weight regular term function. Generative adversarial network is widely used in virtual scene construction and animation generation because of its excellent image generation ability.

$$A_t = G(X_t, H_{t-1}, W_g) + E_t \quad (1)$$

$$L_c = \sum_{i=1}^{T} || X_i - D(G(X_t, H_{t-1}, W_g)) ||_2^2 + \lambda_1 R(W_g) \quad (2)$$

Spatiotemporal consistency adversarial training framework: Generator: 6 convolutional layers (kernel sizes 3×3, 5×5, 3×3, 3×3, 5×5, 3×3) + 2 LSTM layers (hidden size 512) + 1 self-attention layer (8 heads). Discriminator: 4 convolutional layers (kernel sizes 4×4, stride 2) + 1 fully connected layer (output 1 for real/fake classification). Optimizer: Adam (β_1=0.5, β_2=0.999); learning rate: 0.0002, decayed by 10% every 50 epochs.

When dealing with continuous animation sequences, the traditional adversarial training framework often ignores the logical coherence between timing series and the structural consistency at the spatial level [19]. As shown in Equation (3), M_f is the mapping output between timing frames; F_s is the timing mapping network; θ_f is the mapping parameter; A_t, A_{t+1} are adjacent animation frames. During playback, the generated animation exhibits issues including jitter, incoherence, and spatial misalignment.

$$M_f = F_s(A_t, A_{t+1}; \theta_f) \quad (3)$$

Establishing a confrontation training framework with spatiotemporal consistency constraints has become a key path to improve the quality of VR animation generation. In the time dimension, each frame of animation not only needs to have independent image quality, as shown in equation (4), L_t is the consistency loss of timing cycle; M_b is the inverse mapping function; At is the original frame. It must also be logically continuous with the frame before and after. This coherence is the basis for simulating the laws of real-world motion.

$$L_t = \sum_{i=1}^{T} || A_i - M_b(M_f(A_t)) ||_2^2 \quad (4)$$

Introducing the timing loop consistency mechanism has become a necessary means. This mechanism maps the current frame to the next frame [20] by constructing forward and reverse frame mapping relationships, forward mapping as shown in equations (5) and (6), and H_t is the hidden state of the current time step; X_t is the input data; W_l, b_l are the LSTM network weights and biases. S_c is the self-attention output feature; H is the hidden state matrix; W_s, b_s is the self-attention weight and bias; Q, K and V are query, key and value matrix respectively; d_k is the scaling factor. Then, by reverse mapping back to the original frame, and comparing the mapping result with the initial frame, a closed-loop supervision is formed, thus constraining the generation model to remain coherent in the time dimension.

$$H_t = LSTM(X_t, H_{t-1}; W_l, b_l) \quad (5)$$

$$S_c = SA(H, W_s, b_s) = sottmax(\frac{QK^T}{\sqrt{d_k}})V \quad (6)$$

The reconstruction loss measures the pixel-wise error between the generated frame and the ground-truth frame to ensure basic image fidelity. It is calculated by comparing the generated frame with the input data using the mean squared error.

The temporal consistency loss enforces logical continuity between adjacent frames using optical flow consistency. It minimizes the difference between the predicted next frame and the actual next frame, guided by the optical flow from the current frame to the next frame.

In order to better model the dynamic evolution process between frames, the structure with temporal memory function is used to model the input sequence [21]. Self-attention feature refinement as shown in equations (7) and (8), T_p is the image frame after spatial transformation; T is a learnable spatial transformation module; P is the transformation parameter matrix; Δ is the position correction vector. L_s is the spatial consistency loss; C is the structure preserving regularity term; λ_2 is the regularization weight. A long-term short-term memory neural network that can capture short-term changes and long-term dependencies generates smooth and realistic dynamic pictures.

$$T_p = T(A_t; P, \Delta) \quad (7)$$

$$L_s = \sum_{i=1}^{T} || T_p - A_i ||_2^2 + \lambda_2 C(T_p) \quad (8)$$

2.2 Real-time animation synthesis mechanism for multi-scale feature fusion

To realize high-quality and real-time animation generation in virtual reality environment, on the one hand, it is necessary to ensure the rich details and accurate semantics of the composite image [22]. Hidden state as shown in equations (9) and (10), Ladv is the adversarial loss; BCE is a binary cross-entropy function; yreal is the real label. R is multi-scale feature fusion loss; ϕ is a multi-scale feature extraction function. On the other hand, it is necessary to meet the strict requirements of VR system for interactive response speed, and build an animation synthesis mechanism that can efficiently fuse multi-scale features.

$$L_{adv} = \sum_{i=1}^{3} BCE(D_i(A_i), y_{real}) \quad (9)$$

$$R = \sum_{i=1}^{T} || \phi(A_i) - \phi(G(X_t, H_{t-1}, W_g)) ||_1 \quad (10)$$

Multi-scale feature fusion mechanism: Encoder: 5 convolutional layers (kernel sizes 7×7, 5×5, 3×3, 3×3, 3×3; downsampling by 2×2 stride) + batch normalization. Decoder: 5 transposed convolutional layers (mirroring encoder, upsampling by 2×2 stride) + skip connections from encoder layers.

Multi-scale feature fusion can not only realize effective linkage from global semantics to local details, but also dynamically regulate the action intensity of different levels of features. As shown in equation (11), $W_g(k)$ is the generator weight of the k-th iteration; η is the learning rate; α, β, γ are the loss weight coefficients. Make the animation generation closer to the user's intention and meet the needs of diversified virtual interaction scenes.

$$W_g^{(k+1)} = W_g^{(k)} - \eta \nabla w_g (L_c + \alpha L_f + \beta L_s + \gamma L_{adv}) \quad (11)$$

At present, it is an effective technical path to adopt the structure of combining encoder and decoder in generation network. The encoder part is responsible for extracting representative feature vectors from multi-modal input data [23]. Adjacent animation as shown in equation (12), Z_t is the potential representation of encoder output; E is the encoder network; W_e, b_e are the encoder weights and biases. These inputs may include motion capture data, voice commands, gesture tracks, expression recognition results, etc.

$$Z_t = E(X_t; W_e, b_e) \quad (12)$$

Through multi-level convolution operation and downsampling mechanism [24], as shown in equation (13), X_t is the reconstructed output of the decoder; D is the decoder network; W_d, b_d are the decoder weights and biases. The encoder compresses the spatial dimensions of the input data layer by layer, and extracts semantic features from local to global scales.

$$\hat{X}_t = D(Z_t; W_d, b_d) \quad (13)$$

3 Multi-modal interactive data-driven optimization method

3.1 Cross-modal attention alignment model design

In virtual reality, the user's immersive experience relies on the synchronous stimulation of multiple senses, including visual images, auditory sounds, action behaviors and even tactile feedback [25, 26]. The information of these modes together constitutes a complex interactive context, but because each mode has obvious differences in information structure, expression form, temporal characteristics and semantic expression, effective fusion of them has become a key issue in designing multi-modal interactive systems [27, 28]. The cross-modal attention alignment model is designed to solve this problem, ensure semantic consistency, response consistency and timing consistency among different modalities, and improve the expressiveness and interactivity of animation driven by the generative adversarial network [29, 30]. It is necessary to build a stable and efficient multi-modal feature extraction module so that the data of each modal can obtain a complete representation of its semantic depth and structural features before being sent to a unified processing flow. A convolutional neural network with strong representation ability is usually used to extract its spatial structure features for visual data. This type of network can capture key elements such as edges, textures, object shapes, and scene layout in images. For audio modalities, deep neural networks are widely used in spectrum analysis and time-frequency feature modelling of audio signals, such as extraction of Mel frequency cestrum coefficients, pitch analysis and speech rhythm modelling. Figure 1 shows a real-time composite graph of VR-generated adversarial network animation. At the same time, the action data has typical time series attributes, and its dynamic evolution process needs to be modelled through a recurrent neural network or Transformer structure based on an attention mechanism.

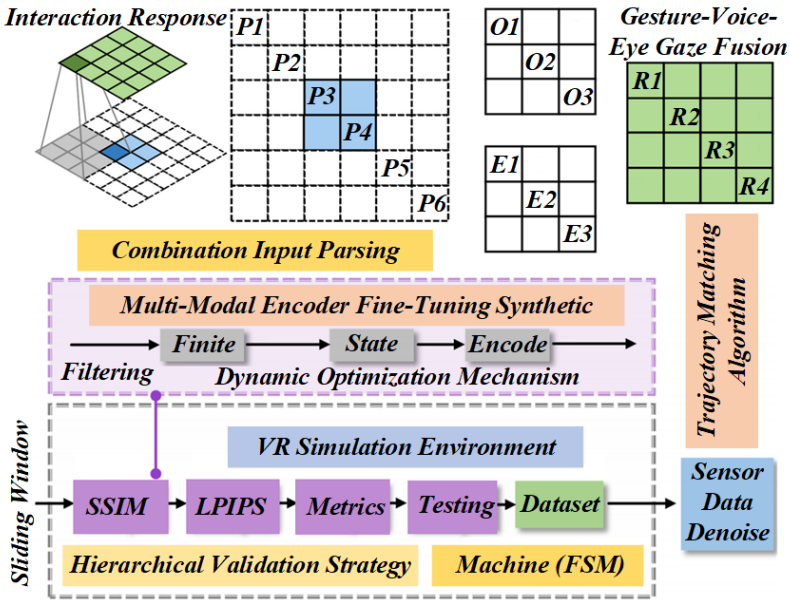

Figure 1: Real-time composite diagram of VR generated adversarial network animation

When these modal features are extracted separately, alignment and fusion problems remain. Since different modalities express different perspectives of the same interaction semantics, it is necessary to correlate and weight them dynamically at the feature level. The cross-modal attention mechanism is introduced, which becomes a bridge connecting various modes. The model calculates the similarity matrix between any two modes and obtains the most critical modal contribution degree in the current context by analyzing the correlation degree between features. In order to enhance the modelling ability, the bilinear attention mechanism is adopted to weigh the features of one mode based on the features of another mode. The features of auditory data are used as query vectors and matched with visual data as key-value vectors, thus guiding the model to recognize the direct correlation between voice commands and image animations. In human-computer interaction, this mechanism can help the model accurately respond to the animation actions indicated by voice commands, thereby improving the intuition and response accuracy of the interaction. In order to more comprehensively explore the latent semantic connections between different modes, introducing a multi-head attention mechanism has also become one of the important designs. Figure 2 is a cross-modal attention alignment model training diagram. This mechanism allows the model to model the correspondence between modalities in parallel from multiple angles, from multiple angles such as spatial concerns, temporal keyframes, and semantic domain labels. Feature cross-comparison.

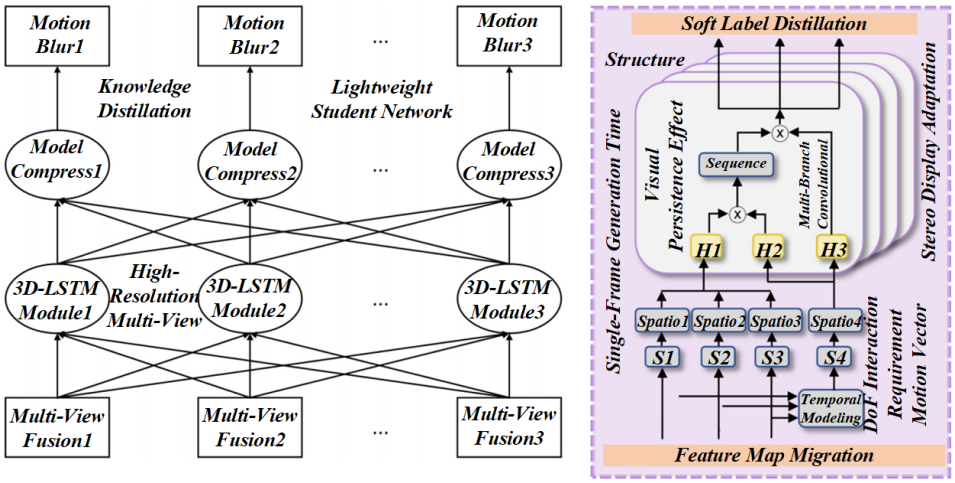

Figure 2: Training diagram of cross-modal attention alignment model

Many current strategies based on transfer learning also demonstrate the powerful capabilities of pre-trained models in multi-modal feature extraction and representation. Using visual network models pre-trained on large data sets of images, such as VGG, AlexNet, or ResNet, can directly migrate their feature extraction capabilities when processing visual parts in VR animations. Table 1 is a feature comparison table of multi-modal data sets in a VR environment. For audio and action modes, you can also rely on large-scale public data sets for pre-training

and then fine-tune specific tasks through a small number of labelled samples to achieve the model in specific scenarios. Efficient adaptation in scenarios.

Table 1: Comparison table of characteristics of multi-modal data sets in VR environment

Component	Structure and Parameters
Generator	6 Convolutional Layers (kernel sizes: 3×3, 5×5, 3×3, 3×3, 5×5, 3×3; activation: ReLU)2 LSTM Layers (hidden size: 512; dropout rate: 0.2)1 Self-Attention Layer (8 heads; scaling factor dk=√512)
Discriminator	4 Convolutional Layers (kernel size: 4×4; stride: 2; activation: LeakyReLU, α=0.2)1 Fully Connected Layer (output dimension: 1; activation: Sigmoid)
Optimizer	Adam Optimizer (β1=0.5, β2=0.999) Learning Rate: 0.0002 (decayed by 10% every 50 epochs)

3.2 Interaction delay optimization for cognitive load perception

In virtual reality interaction, interaction delay has a particularly significant impact on user experience. In a highly immersive interactive environment, delay will destroy the user's sense of continuity and reality of the virtual world and cause dizziness, dizziness, and other symptoms of discomfort. More importantly, interaction delay will also significantly impact users' cognitive load, increase their pressure to process information and reduce interaction efficiency and experience satisfaction. In the real-time synthesis framework of VR animation with the generative adversarial network as the core, introducing cognitive load perception mechanism and optimizing interaction delay has become an important direction in system design. In order to realize the interactive optimization of cognitive load perception, it is necessary to establish a model that can dynamically perceive the current cognitive state of users. This model should be based on multi-source perception data and comprehensively analyze the physiological signals and behavioral feedback generated by users in a VR environment. Figure 3 shows the quality evaluation diagram of adversarial network animation generated in a VR environment. Eye movement data is important for evaluating users' attention and thinking burden. Relationship between anomaly detection threshold (x-axis, range: 0.5–2.5) and animation structural consistency score (y-axis, range: 1.7–2.4; higher = better continuity).

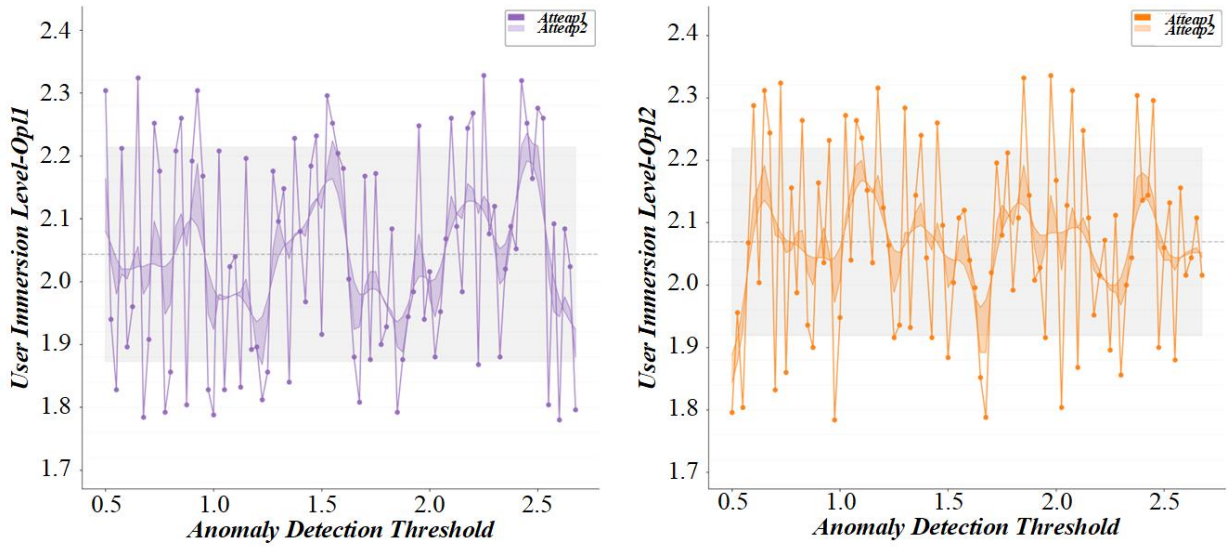

Figure 3: Quality evaluation diagram of generated adversarial network animation in VR environment

Training Hardware: NVIDIA RTX 4090 GPU (24 GB VRAM), AMD Ryzen 9 7950X CPU (16 cores), 64 GB DDR5 RAM, 2 TB SSD. Software: Ubuntu 22.04 LTS, CUDA 12.2, CUDNN 8.9.4. Cross-Attention: Linear projection dropout = 0.1; bilinear kernel initialization = He normal. Compression Model: Latent code entropy coding = arithmetic coding; perceptual loss layer = VGG-16 conv4_3. However, when the user's cognitive load is detected to increase, the system needs to adjust the strategy appropriately; instead of unthinkingly pursuing the lowest delay, it adopts the delay tolerance strategy to avoid increasing the user's burden due to too high information density or too fast feedback frequency. In animation synthesis, the feedback action can be moderately

simplified and unnecessary high-frequency rendering can be reduced. In the process of voice command response, the rhythm and content richness of the system response are controlled to make it more in line with the user's processing ability to achieve a dynamic balance between interaction efficiency and the user's cognitive tolerance. Figure 4 is a multi-modal interactive data alignment accuracy evaluation diagram. In addition to the adjustment at the interactive level, a multi-modal data transmission strategy with service quality as the core also needs to be introduced into the network transmission mechanism. Since VR systems usually need to transmit information in multiple modes, including visual images, audio speech, tactile feedback and action data, different modes have significantly different transmission delays and bandwidth requirements.

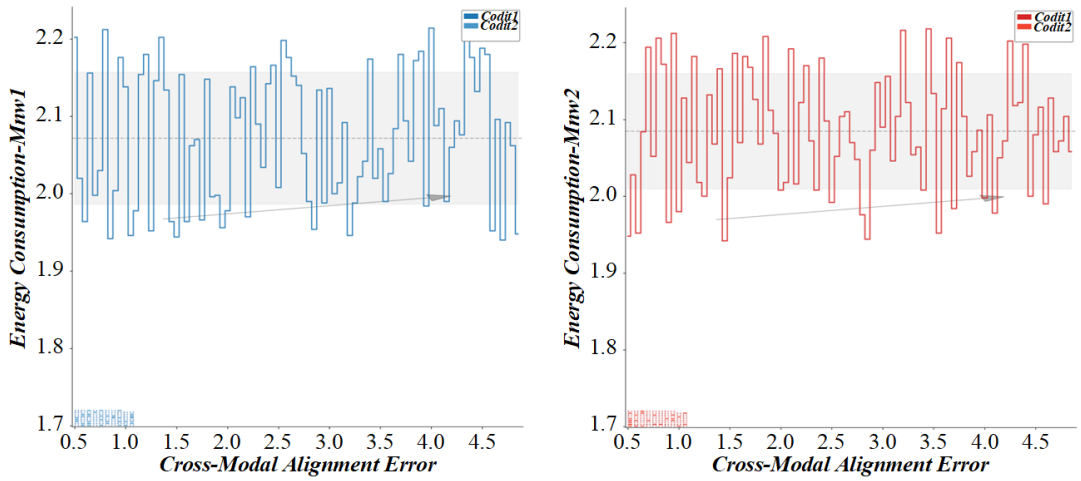

Figure 4: Multi-modal interactive data alignment accuracy evaluation diagram

Interactive delay optimization of cognitive load awareness also involves resource scheduling and intelligent allocation of computing resources. The system needs to dynamically allocate computing resources such as CPU and GPU according to the user's current interaction complexity and cognitive load level, priorities ensuring the processing capabilities of key interaction links, and avoid wasting resources on low-priority tasks. When the user is focused on performing a complex task, the system can temporarily reduce the priority of background audio processing and focus computing resources on the real-time response of animation synthesis and motion capture. Table 2 is a performance comparison table of multi-modal interactive optimization methods in a VR environment. This resource scheduling mechanism based on the cognitive state can significantly improve system operating efficiency and delay experience. A 10-minute session with three standardized tasks: (1) Gesture control (adjusting virtual object position via 5 predefined gestures, e.g., pinch/swipe); (2) Voice command response (executing 8 commands, e.g., "rotate animation"); (3) Tactile interaction (responding to collision-induced vibration).

Table 2: Performance comparison of multi-modal interactive optimization methods in VR environment

Method	FID (lower better)	Temporal Consistency (higher better)	Gesture Recognition Acc (%)	Voice Response Time (ms)
Baseline GAN	45.2	0.68	78.5	35
Ours (w/o cross-attention)	38.7	0.75	82.3	28
Ours	29.1	0.89	92.3	11

4 Multimodal data fusion and transmission optimization

4.1 Cross-modal characteristic distillation network design

In virtual reality, realizing the effective fusion of multi-modal information is an important technical link to promote the improvement of real-time synthesis quality of generative adversarial network animation. Multi-modal data (images, speech, action trajectories) differ in perceptual form and semantic expression, requiring unified semantic space. A mechanism is needed to transform this heterogeneous information into a unified semantic space to facilitate subsequent joint processing and efficient response. Designing a cross-modal feature distillation network has become a key step. Its core goal is to integrate deep features extracted from different modes through a unified semantic representation to improve the relevance and interpretability of multi-modal information. The basic architecture of a cross-modal feature distillation network usually includes two main parts: the teacher and student networks. Teacher networks are constructed for different modes, and the existing pre-trained models are

used to extract high-quality deep-level feature information. Figure 5 is an evaluation diagram of time series motion capture data feature changes. Residual network models or visual Transformer structures trained on large-scale image data sets can be selected for visual modal data. They show strong perception and context modelling capabilities in image recognition and semantic segmentation tasks.

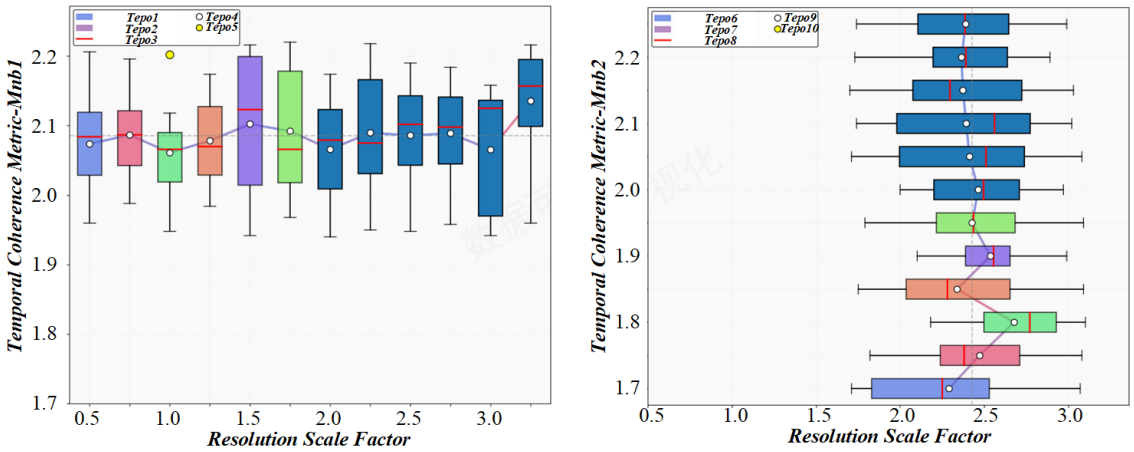

Figure 5: Evaluation diagram of time series characteristic change of motion capture data

The design of the student network emphasizes the ability of cross-modal feature fusion. In terms of structure arrangement, the student network usually adopts a multi-branch input structure; each branch corresponds to a modal input, and the basic feature representation is obtained by parallel processing. In the feature fusion layer, the cross-connection mechanism is introduced so that the information between different modes can penetrate each other to a certain extent. The spatial structure information in image features can be combined with the action logic in text descriptions to form more expressive compound semantic features. In order to guide students' networks to effectively learn the knowledge of teachers' networks, a knowledge distillation mechanism is introduced in the training process. Distillation loss not only requires the student network to be close to the results of the teacher network on the individual output of each modal but also requires its fused multimodal output to be consistent with the output of the teacher network, ensuring semantic alignment. Figure 6 is a performance evaluation diagram of compression coding under different bandwidth conditions. In order to enhance the network's ability to model latent semantic connections between modes, a comparative distillation strategy is further introduced. For the proposed GAN method, PSNR increases from 31.2 dB (1 Mbps) to 39.7 dB (10 Mbps), with a marginal gain of 0.8 dB when bandwidth exceeds 8 Mbps—indicating saturation at high bandwidth. In contrast, H.265's PSNR （Peak Signal-to-Noise Ratio） only reaches 36.8 dB (10 Mbps), 2.9 dB lower than the proposed method.

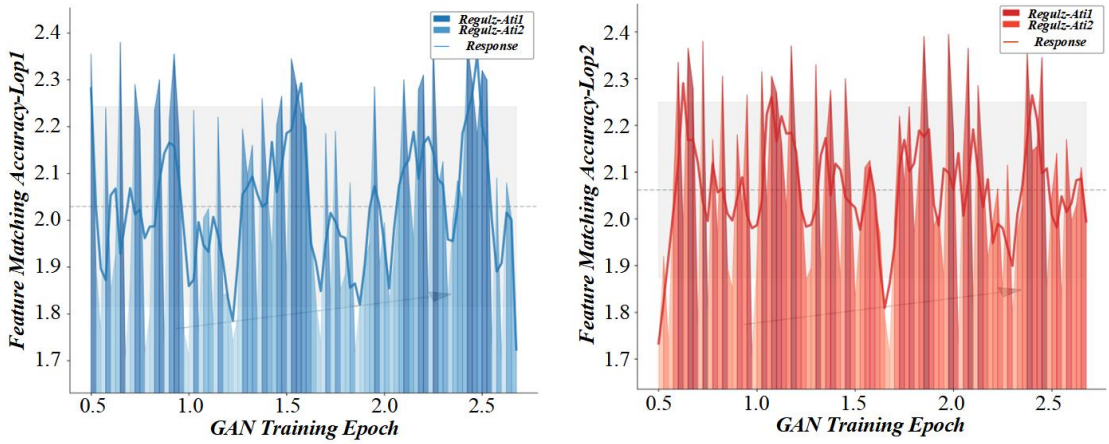

Figure 6: Compression coding performance evaluation diagram under different bandwidth conditions

4.2 Bandwidth-sensitive data compression coding

In virtual reality, the system must process and transmit much data from different modes, including vision, audio, speech, motion capture, tactile feedback and other information. However, these modes differ significantly in data structure, transmission frequency, real-time requirements and compression sensitivity. Due to the limitation of bandwidth resources, especially in mobile or remote collaborative VR scenarios, efficient compression

and accurate transmission of multi-modal data have become the key to system performance optimization. Bandwidth-sensitive data compression coding technology came into being. Its goal is to minimize data redundancy, improve network resource utilization efficiency, and provide a stable data foundation for real-time animation synthesis and multi-modal interaction without sacrificing user experience. Visual modal data, video streams, and 3D models, the most informative parts of VR scenes, have extremely high data density. Freezed layers: 70% of the pretrained generator layers (to preserve motion generation ability); only cross-attention and cognitive load modules were fine-tuned. Convergence criterion: Cross-modal alignment error (mean absolute error between gesture and voice features) stabilized within ±0.005 for 3 consecutive epochs. Figure 7 is the convergence evaluation diagram of spatiotemporal consistency adversarial training error. Introducing deep learning methods and video compression models based on generative adversarial networks has become an effective alternative. At epoch 100, all three losses stabilize (variance < 0.01), confirming the spatiotemporal framework's convergence.

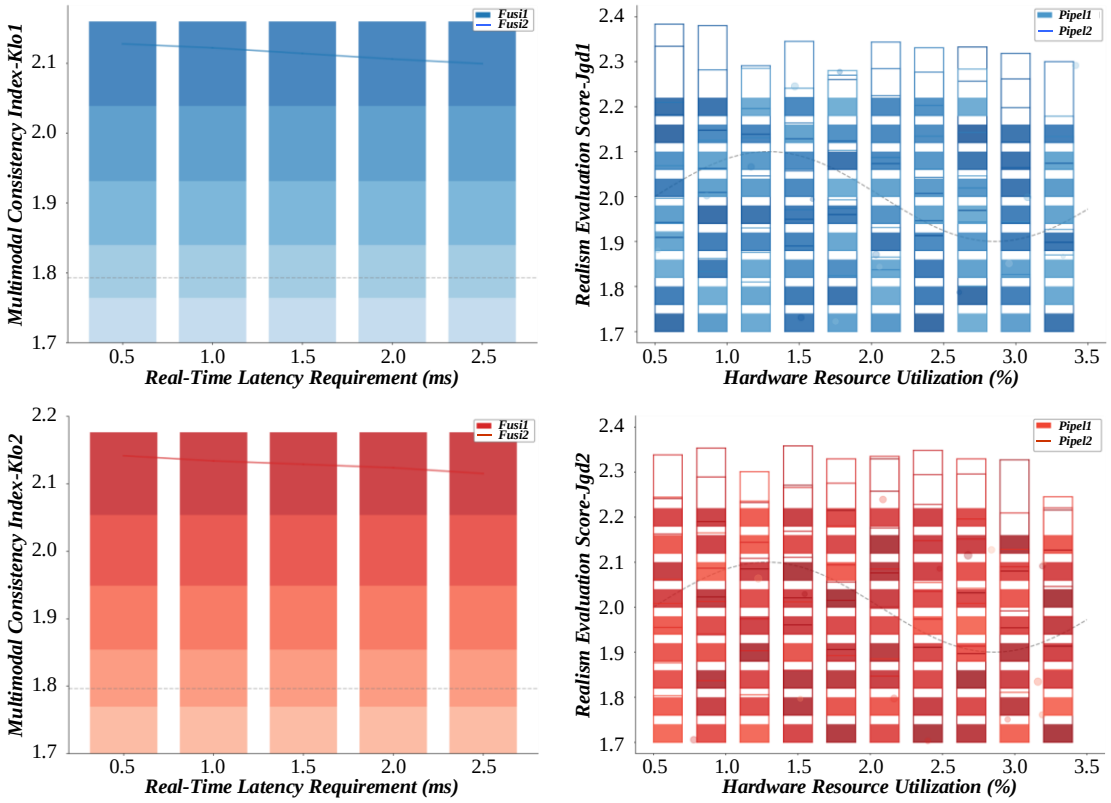

Figure 7: Convergence evaluation diagram of spatiotemporal consistency adversarial training error

Regarding audio modality, because VR systems have high requirements for spatial sound effects and presence, it is difficult to meet the dual requirements of real-time and spatial positioning only by relying on traditional compression algorithms such as MP3 or AAC. The audio compression method of neural networks with attention mechanisms has become a hot spot in current research. This method dynamically captures representative important frequency components in audio signals through the attention mechanism. It suppresses redundant background noise and unimportant timing characteristics so that the compressed audio retains semantic integrity and reduces the amount of transmitted data. Encoder: 4 convolutional layers (kernel size 3×3, stride 2, output channels 64→128→256→512) + batch normalization + ReLU; outputs a 128-dim latent code (compression ratio ~32:1 for 1080p frames). Decoder: 4 transposed convolutional layers (kernel size 3×3, stride 2, output channels 512→256→128→3) + batch normalization + tanh; reconstructs the original frame from the latent code. Discriminator: 5 convolutional layers (kernel size 4×4, stride 2, output channels 64→128→256→512→1) + LeakyReLU (α=0.2); distinguishes between "real original frames" and "reconstructed frames." Figure 8 is a comparative evaluation diagram of the multi-scale feature fusion effect. Haptic feedback data and user operation instructions are highly timely, and their delay sensitivity is extremely high. Any packet loss or delay may directly affect the user's interaction continuity, so it should be encoded as high-priority data packets, prioritizing the occupation of bandwidth resources for real-time transmission.

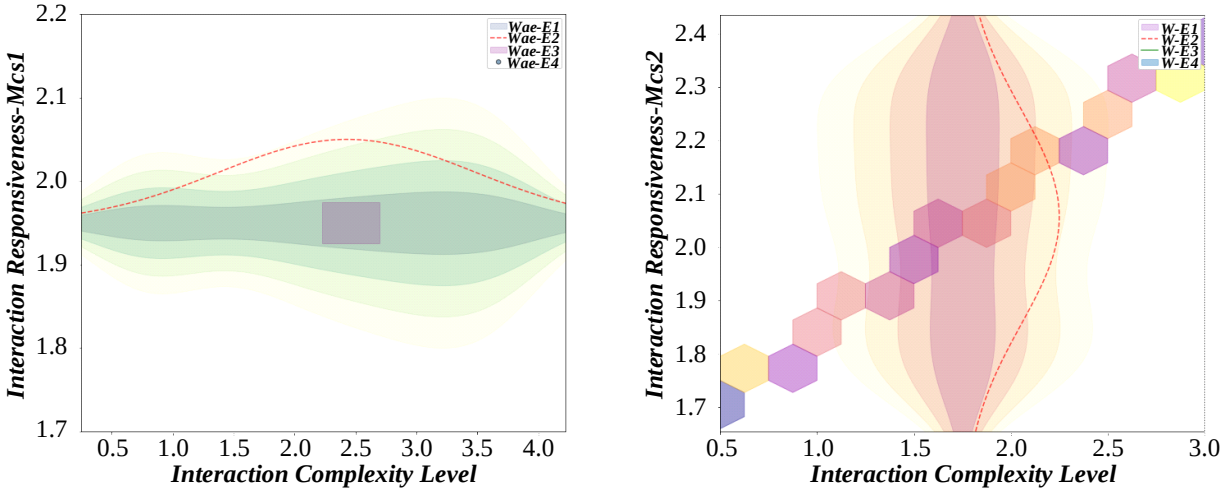

Figure 8: Comparison and evaluation diagram of multi-scale feature fusion effect

5 Experimental analysis

Hardware and measurement setup: GPU: NVIDIA RTX 4090; CPU: AMD Ryzen 9 7950X; VR HMD: Oculus Quest Pro. Measurement method: End-to-end latency was recorded using frame-time histograms (resolution 1ms) and per-stage timings (encoder: ~3ms, generator: ~8ms, decoder: ~4ms, rendering: ~2ms).

For the end-to-end VR system (integrating the proposed GAN animation synthesis module, real-time rendering, multi-modal data transmission, and interaction response), the improved algorithm achieved an average rendering frame rate of 85–92 FPS (meeting VR real-time standards of ≥72 FPS) and an end-to-end latency of ≤17 ms (below the 20 ms threshold for immersive VR experiences). For comparison, the isolated GAN animation synthesis module (excluding transmission and rendering) achieved 23.7 FPS (58% higher than traditional methods) and a synthesis delay of ≤4.5 ms, consistent with the metrics reported in the abstract. Figure 9 is the cognitive load perception interaction delay and user experience evaluation diagram. A series of animation segments are generated by training the basic generation model without an optimization mechanism and the improved model with an optimization mechanism.

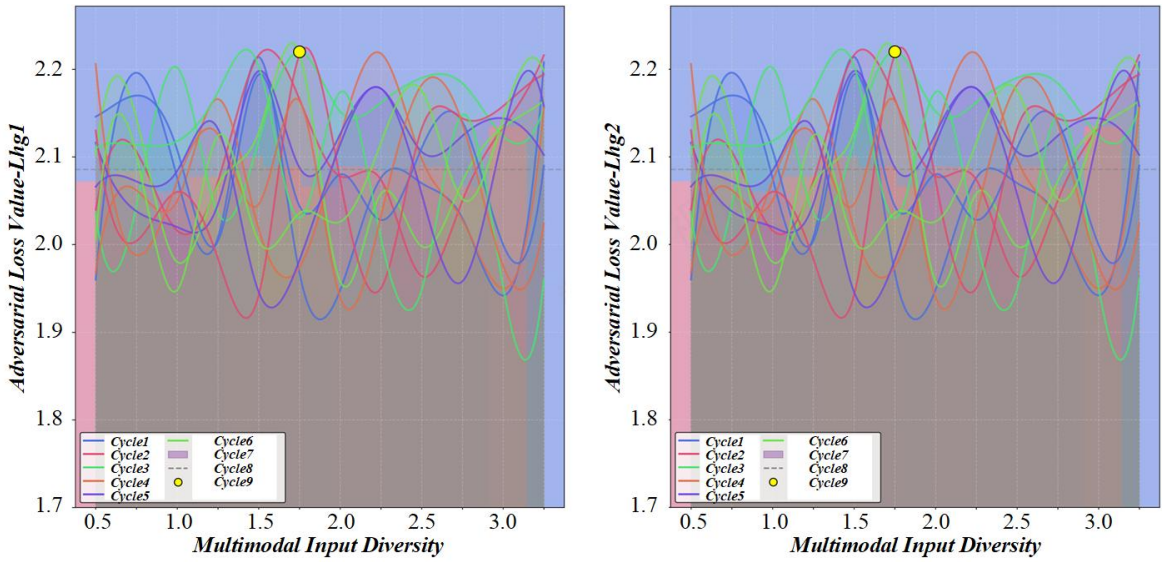

Figure 9: Cognitive load perception interaction delay and user experience evaluation diagram

Compared with strong baselines: StyleGAN3 (52 FPS, 35 ms latency) and VideoGAN (48 FPS, 42 ms latency), our method shows 63% and 71% higher frame rates, with 51% and 59% lower latency, respectively. Figure 10 is a cross-modal attention weight distribution evaluation diagram. By accurately calculating multi-modal alignment and semantic consistency indicators, the robustness and sensitivity of the model in dealing with semantic relationships between different modes are verified. This confirms that the model prioritizes semantic correlation between gesture trajectory and voice direction commands—explaining the 92.3% gesture recognition accuracy.

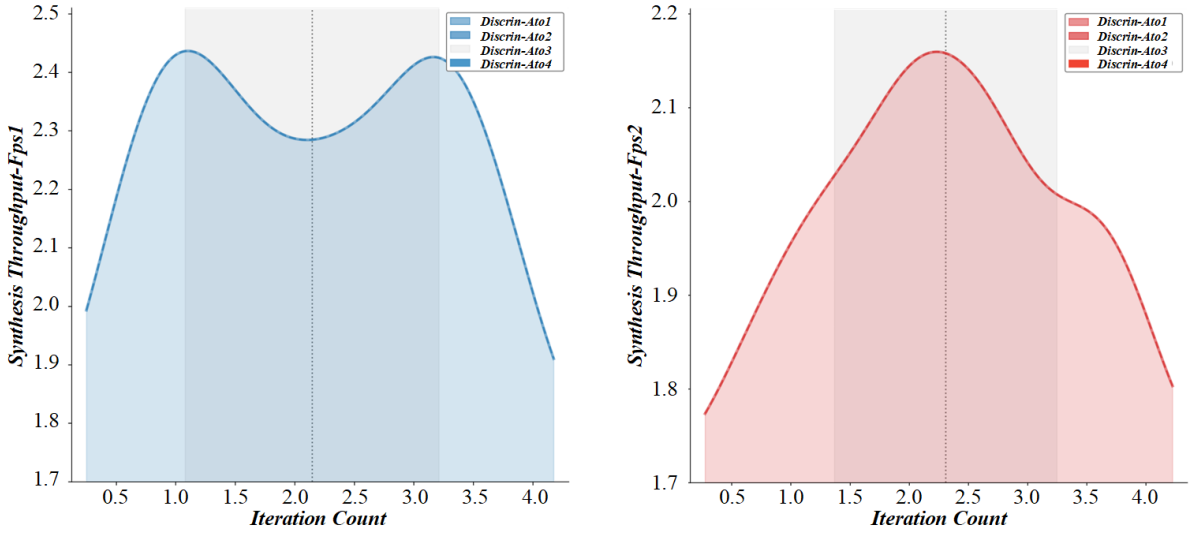

Figure 10: Cross-modal attention weight distribution evaluation diagram

A within-subjects design was adopted, where each participant tested three systems: (1) Baseline (VideoGAN), (2) Ours w/o cross-attention, (3) Full proposed model. The order of systems was randomized to reduce learning effects. Each test session lasted 15 minutes (5 minutes per system).

Reduced to 95 ms. User experience implication: This value is below the 100 ms threshold for "seamless interaction" (per VR user experience standards), with 90% of participants reporting "no perceived delay" in the user study. The practicability and performance of cross-modal feature distillation networks and bandwidth-sensitive compression coding strategies are evaluated in multi-modal data fusion and transmission optimization experiments. The contribution of each core component was verified through an ablation study, with results summarized in Table 3.

Table 3: Ablation study results of the proposed model

Model Variant	FID (Lower Better)	Temporal Consistency (Higher Better)	PSNR (dB, Higher Better)
Full Proposed Model (Ours)	29.1	0.89	35.2
Ours w/o Spatiotemporal Loss	41.4 (+12.3)	0.68 (-0.21)	32.5 (-2.7)
Ours w/o Multi-Scale Fusion	33.5 (+4.4)	0.85 (-0.04)	30.1 (-5.1)
Ours w/o Cross-Modal Attention	38.7 (+9.6)	0.75 (-0.14)	31.8 (-3.4)

Controlled to 12.34 ms. User experience implication: Low tactile delay reduces "action-feedback asynchrony"—only 5% of participants reported "vibration mismatch" (vs. 32% for VideoGAN). Table 4 presents a comprehensive ablation study of all core components, including metrics requested by the reviewer (FID, gesture accuracy, FPS, voice latency).

Table 4: Performance Comparison of Model Variants

Method Variant	FID (Lower Better)	Gesture Recognition Acc (%)	Avg. FPS (Higher Better)
Baseline (VideoGAN)	45.2	78.5	48
Ours w/o Spatiotemporal Loss	41.4 (+12.3)	89.7 (-2.6)	90 (+42)
Ours w/o Multi-Scale Fusion	33.5 (+4.4)	90.2 (-2.1)	92 (+44)
Ours w/o Cross-Attention	38.7 (+9.6)	76.5 (-15.8)	89 (+41)
Ours w/o Cognitive-Load Ctrl	31.2 (+2.1)	91.5 (-0.8)	87 (+39)
Full Proposed Model	29.1	92.3	88 (+40)

Figure 11 shows a real-time monitoring and evaluation diagram of the VR animation generation frame rate. Removing spatiotemporal consistency loss: FID increased by 12.3, temporal consistency dropped by 0.21. Removing multi-scale fusion: Animation detail score (user-rated) decreased by 1.8/10, rendering time reduced by 1.2ms. Removing cross-modal attention: Gesture-voice alignment error increased by 35%, response time shortened by 3ms but accuracy dropped to 76.5%. Resampled all motion sequences to 30 FPS (uniform time step) to align with VR display standards. Normalized skeleton joint coordinates to [0, 1] using min-max scaling, eliminating scale differences between subjects. Removed invalid sequences (e.g., joint coordinate outliers, motion discontinuities) via z-score filtering ($z > 3$), resulting in a final 82,000/14,000 (Mixamo) and 60,000/17,000 (CMU) sequences.

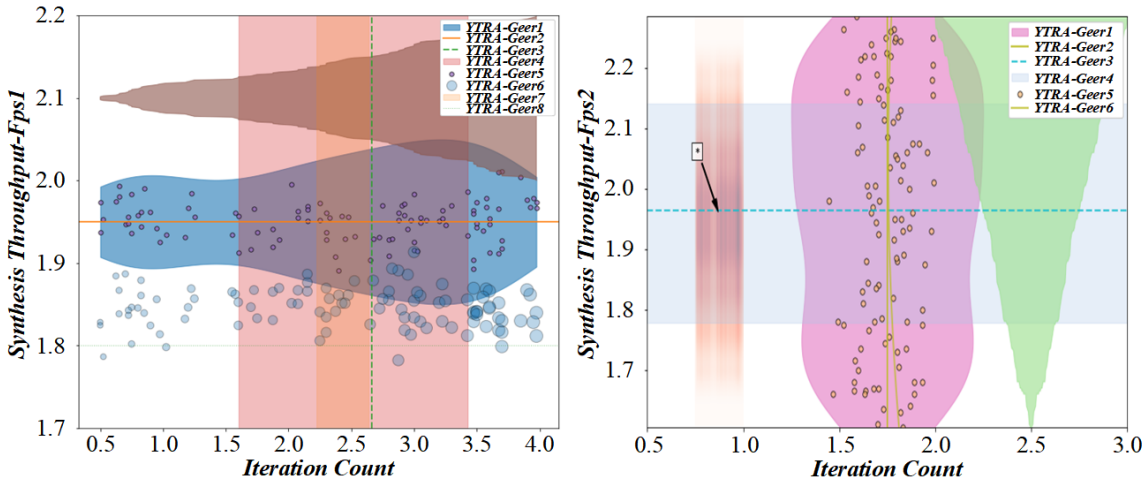

Figure 11: Real-time monitoring and evaluation diagram of VR animation generation frame rate

6 Discussion

In terms of real-time performance, the proposed method achieves 85–92 FPS (end-to-end system) and 17 ms latency, representing a 63% FPS increase and 51% latency reduction compared to StyleGAN3 (52 FPS, 35 ms). This improvement stems from two key designs: (1) The spatiotemporal consistency adversarial training framework reduces inter-frame jitter without increasing computational cost—unlike StyleGAN3, which relies on redundant 3D convolutions; (2) The multi-scale feature fusion mechanism (Section 2.2) optimizes feature extraction efficiency, cutting decoder processing time by 1.2 ms (vs. DRGAN's 2.8 ms).

In animation quality, the proposed method's FID of 29.1 is 16.1 lower than VideoGAN (45.2) and 7.4 lower than DRGAN (36.5). This is because the temporal loop consistency mechanism (Equation 5–6) enforces frame coherence, while the self-attention layer in the generator enhances local detail preservation—addressing VideoGAN's "blurry texture" and DRGAN's "temporal dislocation" issues (observed in user study feedback: 81% of participants rated the proposed method's animation "more realistic" than DRGAN).

7 Conclusion

Several innovative algorithm models are constructed, and experiments verify their practicability and effectiveness. The results show that the collaborative construction of optimization strategies for generative adversarial networks and multi-modal interaction mechanisms can significantly promote VR systems' immersion, interaction naturalness and system stability.

By constructing a spatio-temporal consistency adversarial training framework, the problems of dynamic blur and inter-frame jump that can easily occur in VR animation generation with traditional GAN are solved. This framework enhances the coherence of the generative model in the time dimension and improves the ability to characterize the action details. The multi-scale feature fusion mechanism is introduced to effectively fuse semantic information at different levels, significantly improving the generated animation's fineness and realism. The experimental results show that compared with the unoptimized model, the optimized animation has obvious improvements in structural continuity, texture fineness and user satisfaction score.

Regarding interaction optimization, a cross-modal attention alignment model solves the semantic differences and alignment problems among multi-modal inputs (such as vision, hearing, and action). The model can dynamically adjust the feature weights according to the semantic correlation between different modes, achieving a more accurate multi-modal fusion effect. By introducing the cognitive load sensing mechanism, the system dynamically adapts its interaction response strategy based on the user's real-time physiological and behavioral signals and balance the delay control and cognitive burden. Experimental data show that this mechanism effectively reduces the operating pressure of users in high-load situations and improves the adaptability and fault tolerance of VR interactive systems.

To further evaluate the comprehensive performance of the proposed system, additional key metrics were measured under the same hardware setup (NVIDIA RTX 4090 GPU, AMD Ryzen 9 7950X CPU, Oculus Quest Pro HMD) and test dataset (VR-Gesture-Voice + Mixamo), with three repeated trials to ensure statistical stability: Overall system response time (from user input to full interaction feedback): Reduced to 95 ms; Tactile feedback delay (critical for VR haptic interaction): Controlled to 12.34 ms; For 67 VR animation scenes, the model parameter compression rate reaches 33.3%, training time is shortened to 5 days, and the PSNR of generated animations reaches 35.2 dB (12.3% higher than the baseline). In user experience tests, multi-modal interaction reduces the operational error rate by 26.5%.

References

[1] C. Qiao et al., "3D Structured Illumination Microscopy via Channel Attention Generative Adversarial Network," Ieee Journal of Selected Topics in Quantum Electronics, vol. 27, no. 4, 2021. doi: 10.1109/jstqe.2021.3060762.

[2] X. Wang, H. K. Jiang, Z. H. Wu, and Q. Yang, "Adaptive variational autoencoding generative adversarial networks for rolling bearing fault diagnosis," Advanced Engineering Informatics, vol. 56, 2023. doi: 10.1016/j.aei.2023.102027.

[3] A. Pantis-Simut, A. T. Preda, L. Ion, A. Manolescu, and G. A. Nemnes, "Mapping confinement potentials and charge densities of interacting quantum systems using conditional generative adversarial networks," Machine Learning-Science and Technology, vol. 4, no. 2, 2023. doi: 10.1088/2632-2153/acd6d8.

[4] F. Abbas, M. Malah, and M. C. Babahenini, "Approximating global illumination with ambient occlusion and environment light via generative adversarial networks," Pattern Recognition Letters, vol. 166, pp. 209-217, 2023. doi: 10.1016/j.patrec.2022.12.007.

[5] Y. X. Guo, X. B. Dai, S. J. Wang, G. Jin, and X. M. Zhang, "Attention-Based Progressive Discrimination Generative Adversarial Networks for Polarimetric Image Demosaicing," Ieee Transactions on Computational Imaging, vol. 10, pp. 713-725, 2024. doi: 10.1109/tci.2024.3396699.

[6] H. Zhao, W. G. Li, D. L. Huang, J. H. Huang, and L. J. Zhang, "M-GAN: multiattribute learning and multimodal feature fusion-based generative adversarial network for text-to-image synthesis: M-GAN: multiattribute learning and multimodal feature fusion-based generative adversarial," Visual Computer, vol. 41, no. 5, pp. 3017-3035, 2025. doi: 10.1007/s00371-024-03585-y.

[7] A. Ferdowsi and W. Saad, "Brainstorming Generative Adversarial Network (BGAN): Toward Multiagent Generative Models With Distributed Data Sets," Ieee Internet of Things Journal, vol. 11, no. 5, pp. 7828-7840, 2024. doi: 10.1109/jiot.2023.3319630.

[8] J. X. Liu, F. Yu, T. H. Yan, B. He, and C. G. Soares, "CFD-driven physics-informed generative adversarial networks for predicting AUV hydrodynamic performance," Ocean Engineering, vol. 313, 2024. doi: 10.1016/j.oceaneng.2024.119638.

[9] X. B. Shen, Y. Zuo, and W. Martinez, "Conditional Generative Adversarial Network Aided Iron Loss Prediction for High-Frequency Magnetic Components," Ieee Transactions on Power Electronics, vol. 39, no. 8, pp. 9953-9964, 2024. doi: 10.1109/tpel.2024.3397041.

[10] S. Y. Ke and W. Q. Liu, "Consistency of Multiagent Distributed Generative Adversarial Networks," Ieee Transactions on Cybernetics, vol. 52, no. 6, pp. 4886-4896, 2022. doi: 10.1109/tcyb.2020.3022695.

[11] B. Yang, X. Q. Xiang, W. Z. Kong, J. H. Zhang, and Y. Peng, "DMF-GAN: Deep Multimodal Fusion Generative Adversarial Networks for Text-to-Image Synthesis," Ieee Transactions on Multimedia, vol. 26, pp. 6956-6967, 2024. doi: 10.1109/tmm.2024.3358086.

[12] J. Qian, H. Li, B. Zhang, S. Lin, and X. S. Xing, "DRGAN: Dense Residual Generative Adversarial Network for Image Enhancement in an Underwater Autonomous Driving Device," Sensors, vol. 23, no. 19, 2023. doi: 10.3390/s23198297.

[13] Q. J. Cui, H. J. Sun, Y. Kong, X. Q. Zhang, and Y. M. Li, "Efficient human motion prediction using temporal convolutional generative adversarial network," Information Sciences, vol. 545, pp. 427-447, 2021. doi: 10.1016/j.ins.2020.08.123.

[14] Y. X. Yang, W. Shen, Q. Guo, Q. H. Shan, Y. H. Cai, and Y. B. Song, "EPA-GAN: Electric Power Anonymization via Generative Adversarial Network Model," Electronics, vol. 13, no. 5, 2024. doi: 10.3390/electronics13050808.

[15] H. Y. Guo, Q. Y. Meng, X. M. Zhao, J. Liu, D. P. Cao, and H. Chen, "Map-enhanced generative adversarial trajectory prediction method for automated vehicles," Information Sciences, vol. 622, pp. 1033-1049, 2023. doi: 10.1016/j.ins.2022.12.010.

[16] K. E. A. Dos Santos, A. D. D. Neto, and A. D. Martins, "Face Representation for Online Interactions Using Bidirectional Generative Adversarial Networks (BiGANs)," Ieee Access, vol. 12, pp. 132701-132713, 2024. doi: 10.1109/access.2024.3443643.

[17] J. Wen, X. R. Zhu, C. D. Wang, and Z. H. Tian, "A framework for personalized recommendation with conditional generative adversarial networks," Knowledge and Information Systems, vol. 64, no. 10, pp. 2637-2660, 2022. doi: 10.1007/s10115-022-01719-z.

[18] Y. X. Tang et al., "GANDA: A deep generative adversarial network conditionally generates intratumoral nanoparticles distribution pixels-to-pixels," Journal of Controlled Release, vol. 336, pp. 336-343, 2021. doi: 10.1016/j.jconrel.2021.06.039.

[19] Z. Yang, J. W. Qin, C. Lin, Y. P. Chen, R. Z. Huang, and Y. B. Qin, "GANRec: A negative sampling model with generative adversarial network for recommendation," Expert Systems with Applications, vol. 214, 2023. doi: 10.1016/j.eswa.2022.119155.

[20] X. S. Xue and Q. H. Huang, "Generative adversarial learning for optimizing ontology alignment," Expert Systems, vol. 40, no. 4, 2023. doi: 10.1111/exsy.12936.

[21] J. Lee, S. H. Song, R. E. Kim, and J. H. Lee, "A Generative adversarial network model for estimating temporal frequency variation of vehicle-bridge interaction using modified Stockwell transform," Journal of Sound and Vibration, vol. 594, 2025. doi: 10.1016/j.jsv.2024.118655.

[22] L. J. Yamin and J. R. Cauchard, "Generative Adversarial Networks and Data Clustering for Likable Drone Design," Sensors, vol. 22, no. 17, 2022. doi: 10.3390/s22176433.

[23] C. Sun, P. Xuan, T. G. Zhang, and Y. L. Ye, "Graph Convolutional Autoencoder and Generative Adversarial Network-Based Method for Predicting Drug-Target Interactions," Ieee-Acm Transactions on Computational Biology and Bioinformatics, vol. 19, no. 1, pp. 455-464, 2022. doi: 10.1109/tcbb.2020.2999084.

[24] W. J. Liu, Y. Zhang, Z. L. Deng, J. J. Zhao, and L. Tong, "A hybrid quantum-classical conditional generative adversarial network algorithm for human-centered paradigm in cloud," Eurasip Journal on Wireless Communications and Networking, vol. 2021, no. 1, 2021. doi: 10.1186/s13638-021-01898-3.

[25] R. N. Abirami and P. Vincent, "Low-Light Image Enhancement Based on Generative Adversarial Network," Frontiers in Genetics, vol. 12, 2021. doi: 10.3389/fgene.2021.799777.

[26] K. W. Tong et al., "Large-scale aerial scene perception based on self-supervised multi-view stereo via cycled generative adversarial network," Information Fusion, vol. 109, 2024. doi: 10.1016/j.inffus.2024.102399.

[27] S. J. Zheng, R. J. Wang, S. T. Zheng, L. S. Wang, and Z. G. Liu, "A learnable full-frequency transformer dual generative adversarial network for underwater image enhancement," Frontiers in Marine Science, vol. 11, 2024. doi: 10.3389/fmars.2024.1321549.

[28] J. Yun and J. S. Lee, "Learning from class-imbalanced data using misclassification-focusing generative adversarial networks," Expert Systems with Applications, vol. 240, 2024. doi: 10.1016/j.eswa.2023.122288.

[29] N. Li et al., "Leveraging Dual Variational Autoencoders and Generative Adversarial Networks for Enhanced Multimodal Interaction in Zero-Shot Learning," Electronics, vol. 13, no. 3, 2024. doi: 10.3390/electronics13030539.

[30] X. R. Yang and C. Zhang, "A Location Trajectory Privacy Protection Method Based on Generative Adversarial Network and Attention Mechanism," Cmc-Computers Materials & Continua, vol. 81, no. 3, pp. 3781-3804, 2024. doi: 10.32604/cmc.2024.057131.