Recognition and Modeling of User Interaction Behaviors in Virtual Reality Based on 3D Convolutional Neural Networks

Huiling Wu
Digital Media Teaching and Research Office, Qijiang Campus, Big Data College, Chongqing Yitong University, Chongqing 401420, China
Corresponding author: Huiling Wu (e-mail: m13637877731@163.com)

ABSTRACT Accurate recognition of continuous user interactions in virtual reality (VR) is essential for intelligent perception systems and provides methodological support for multimodal information processing in electromagnetic sensing and nextgeneration human–machine interaction. However, fragmented action sequences, weak long-range dependencies, and semantic discontinuities remain major challenges in complex immersive environments. This study proposes a crosssegment spatiotemporal attention and behavior pattern modeling framework based on a three-dimensional convolutional neural network (3D CNN). The framework integrates multi-layer 3D convolution with residual learning for robust feature extraction, employs cross-segment spatiotemporal attention and Bidirectional Long Short-Term Memory (BiLSTM) networks to capture long-range temporal dependencies, and introduces graph neural networks with boundary smoothing and graph consistency constraints to preserve behavioral structure and action continuity. Experimental results demonstrate an average recognition accuracy of 87.2% for complex actions, a behavior pattern consistency of 95.2%, and real-time inference performance of 39.2 FPS while maintaining superior robustness under occlusion and motion blur. Beyond VR interaction analysis, the proposed hierarchical spatiotemporal representation and structured modeling strategy provides useful insights for adaptive signal interpretation, multimodal perception, and intelligent decisionmaking in electromagnetic sensing and wireless interactive systems.

INDEX TERMS virtual reality, 3D convolutional neural network, spatiotemporal attention, graph neural network, electromagnetic sensing

I. INTRODUCTION

THE development of VR technology has made immersive interaction a core area of human-computer interaction research. Continuous user interaction in a virtual environment carries information about operational intentions and behavioral patterns, directly impacting user experience and task efficiency. This is crucial for industries such as engineering, as VR is used to simulate complex mechanical operations and advanced fabric handling technologies. High-precision identification of continuous interactive behaviors and their structured patterns provide a foundation for decision-making in intelligent interactive systems, improving the naturalness and adaptability of interactions [1,2]. Addressing fragmented action sequences and incomplete semantics is crucial for intelligent analysis and human-computer collaboration in complex interactive scenarios, providing technical support for applications such as immersive training, education, and medical rehabilitation [3,4].

Current research attempts to extract spatiotemporal features using three-dimensional convolutional neural networks, combining residual connections and batch normalization to maintain feature stability. Temporal convolutional networks or bidirectional long short-term memory networks are used to capture long-range dependencies in action sequences, and multi-stage convolution or attention mechanisms are employed to mitigate over-segmentation. These methods have achieved some success in video action recognition and short-term action classification tasks. However, in continuous VR interaction, fragmented action sequences, insufficient capture of long-range dependencies, and structured representation of behavioral patterns remain unresolved, limiting action recognition accuracy and understanding of interactive behaviors [5-7]. Bidirectional long short-term memory networks process temporal signals through recurrent state transitions, while the graph structure preserves fixed topological relations across distant segments, maintaining dependencies at the structural level.

To address the fragmented and incomplete semantics

of continuous interactive behavior recognition in VR, this paper proposes a cross-segment spatiotemporal attention and interactive behavior pattern graph modeling method based on a 3D convolutional neural network. Behavior patterns refer to structured sequences of multiple actions with temporal and spatial dependencies, emphasizing the organizational regularity and logical relationships between actions rather than simple stacking of individual actions. In its implementation, a multi-layer 3D convolutional network is first constructed. The output of each convolutional layer undergoes batch normalization and residual connections to maintain spatial feature stability. The temporal correlation matrix of each segment is then calculated, and keyframe and continuous action features are weighted and aggregated. Long-range temporal dependencies are extracted using a multi- layer bidirectional long short-term memory network. Interactive action categories are mapped using fully connected layers, and action boundary smoothing losses are used to constrain sequence continuity. On this basis, action sequence nodes are represented as action category features, and edges represent the spatiotemporal dependencies between actions. By constructing a structured behavior pattern graph through a graph neural network (GNN) and jointly optimizing classification, smoothing, and graph consistency loss during end-to-end training, the modeling method achieves the unification of action recognition and behavior pattern modeling. This improves the integrity of action sequences and the system analysis capability in complex interactive scenarios. This modeling method is of great value for accurately assessing and optimizing the procedural skills required in complex manufacturing processes.

II. RELATED WORK

In VR, the recognition and modeling of user interaction actions have become an important direction for improving the interactive experience and system intelligence. Raimbaud et al. proposed a task-centered method to design and evaluate the interaction process, and compared the usability differences of different evaluation methods in the case of hazard identification. This study emphasized the impact of evaluation strategies on the accuracy of action interpretation, thus highlighting the necessity of modeling method design [8]. While improving the recognition and interactive experience, related studies also focus on the efficiency and accuracy issues in the interaction modeling process. Gan et al. combined deep image three-dimensional modeling with hybrid intelligent collision detection algorithm to build a high-precision interaction system. Its improvement in modeling and detection efficiency reflected the important value of algorithm optimization for virtual environment interaction modeling [9]. Overall, these studies have made progress in action recognition, system modeling, and interaction evaluation, but there are still deficiencies in solving the problems of continuous action segmentation and incomplete semantics [10-12].

In the study of spatiotemporal feature modeling, 3D convolutional neural networks have demonstrated powerful feature extraction and representation capabilities. Li et al. proposed the Spatio-Temporal Adaptive 3D Convolutional Neural Network, which achieved multi-level modeling of traffic flow data through an adaptive 3D convolution module and an external factor processing module. This method showed that adaptive convolution had high flexibility and expressiveness in complex dynamic environments [13]. Althou gh these methods have demonstrated excellent performance in traffic prediction, anomaly detection, and medical imaging, they still face shortcomings in the application of continuous interactive action recognition and pattern modeling in VR [14-16].

Existing research primarily focuses on short-duration actions or single-task scenarios. During continuous interactions, action boundaries are often frequently cut, resulting in the loss of long-range dependency information. Deep convolution and recurrent structures improve recognition accuracy, but they struggle to avoid temporal shifts caused by occlusion and noise in immersive environments. Attention mechanisms excel at extracting keyframes, but they lack the ability to maintain the overall coherence of action sequences. Their overreliance on local feature weighting limits long-term continuity. Graph structure modeling methods enhance the ability to express interactive relationships, but lack boundary smoothing and structural consistency constraints. Prediction results are prone to fragmentation under complex interactions, limiting the ability to recover complete semantics. These shortcomings indicate that while existing methods have improved recognition and modeling capabilities across various dimensions, they still lack a robust solution for continuous action recognition and structured representation of behavioral patterns in VR.

III. USER INTERACTION BEHAVIOR MODELING METHODS

The overall framework of the user interaction behavior modeling method is shown in Figure 1. The framework includes five components: input, feature extraction, representation modeling, pattern modeling, and output. It implements the complete process of data processing, feature extraction, attention aggregation, sequence modeling, and classification prediction layer by layer.

In Figure 1, at the feature extraction layer, 3D convolution and pooling structures are used to capture local action information across frames. The application of residual paths mitigates gradient decay and distribution drift while maintaining the expressive power of deep features, making the extracted representation more stable. The representation modeling layer uses a spatiotemporal attention mechanism to capture cross- timestep correlations. These two components are combined in a cross-segment correlation matrix to form a global representation. At the pattern modeling layer, a bidirectional architecture contextualizes temporal features. Combined with multimodal fusion, action representations are uniformly embedded, ensuring the complete transfer of

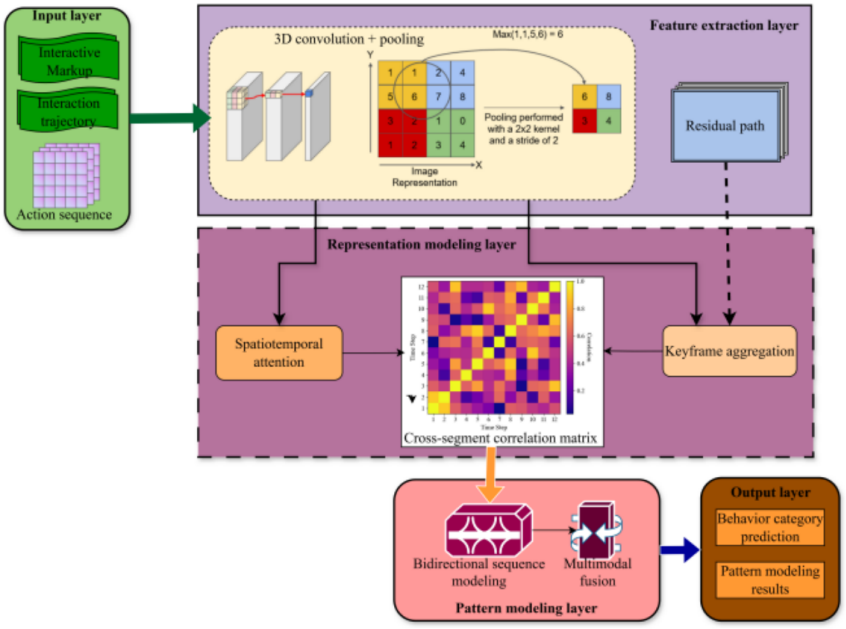

FIGURE 1. Overall framework for user interaction behavior modeling

interaction semantics. The final output layer predicts action categories and provides pattern modeling results to support subsequent analysis.

A. DATA PREPROCESSING AND FEATURE EXTRACTION

Raw VR interaction data are recorded as continuous frame sequences. To avoid timing offset caused by uneven sampling between frames, the sequence is first resampled to a uniform frame rate, and the coordinates of the pose points are zero-mean normalized in the spatial dimension to eliminate the scale effect caused by individual differences. The processed action sequence is divided into fixed-length segments along the temporal axis, where each segment corresponds to a predefined number of consecutive frames and serves as the basic input unit for subsequent three-dimensional convolution operations [17,18]. For sequences whose total frame length exceeds the segment length, segmentation is performed in a sliding manner without altering the original temporal order, while shorter sequences are zero-padded to match the fixed segment length.

After the segment input, it is subjected to feature extraction by a three-dimensional convolutional network. Each convolution block applies a fixed 3D kernel followed by batch normalization and a residual projection that aligns input and output dimensions to maintain stable feature propagation. The residual projection uses a linear mapping so that temporal and spatial resolutions remain consistent across stacked layers. It is assumed that the input action segment tensor is $X \in RT \times H \times W \times C$, where T represents the number of time steps, H and W represent the spatial dimensions, and C denotes the number of input channels [19,20]. The calculation process of the three-dimensional convolution layer is defined as:

$$X_{t,h,w,c'}^{(l)} = \sum_{c=1}^{C} \sum_{\tau=0}^{k_t-1} \sum_{i=0}^{k_h-1} \sum_{j=0}^{k_w-1} W_{c',c,\tau,i,j}^{(l)} \cdot X_{t+\tau,h+i,w+j,c}^{(l-1)} + b_{c'}^{(l)} \tag{1}$$

In Formula (1), $X_{t,h,w,c}^{(l-1)}$ denotes the input feature at temporal index t, spatial location (h, w), and input channel c from the $(l-1)$-th layer, while $X_{t,h,w,c'}^{(l)}$ denotes the output feature at output channel c' of the l-th layer. $W_{c',c,\tau,i,j}^{(l)}$ represents the three-dimensional convolution kernel connecting input channel c to output channel c' with temporal offset τ and spatial offsets (i, j), and $b_{c'}^{(l)}$ denotes the bias term associated with the c'-th output channel. The kernel size is (k_t, k_h, k_w). This formulation explicitly models channel-wise aggregation and local spatiotemporal interactions during feature extraction. To avoid gradient vanishing and feature distribution drift in deep convolution, the convolution output is normalized by a batch normalization layer:

$$\hat{F}^{(l)} = \frac{F^{(l)} - \mu^{(l)}}{\sqrt{(\sigma^{(l)})^2 + \epsilon}} \tag{2}$$

In Formula (2), $\mu^{(l)}$ and $\sigma^{(l)}$ represent the mean and standard deviation of the l-th layer features, respectively, and ϵ is a numerical stability constant. The normalized features are added to the input of the previous layer through a residual connection to ensure smooth information flow during deep stacking and avoid weakening important spatial structures. Figure 2 shows how the residual structure is embedded in different convolutional layers.

In Figure 2, the input action sequence is input into the three-dimensional convolution layer in the form of a

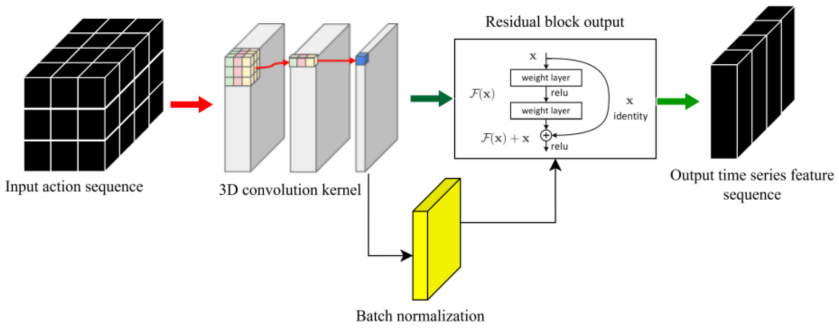

FIGURE 2. Schematic diagram of the interactive behavior recognition network structure based on 3D convolution

tensor, and local features are extracted in the spatiotemporal dimension. The convolution output is constrained by the batch normalization layer and fused with the input features through the residual path to ensure the stability and integrity of the information during deep transmission. It is finally mapped into a unified high-dimensional temporal feature sequence, providing input for subsequent spatiotemporal attention modeling.

After the convolution and residual structure are stacked, each time step is mapped into a high-dimensional feature vector sequence. It is assumed that the l-th layer output sequence is:

$$Z^{(l)} = \{z_1^{(l)}, z_2^{(l)}, \ldots, z_T^{(l)}\}, \quad z_t^{(l)} \in R^d \tag{3}$$

In Formula (3), d is the feature dimension. This representation ensures that the input of the subsequent spatiotemporal attention mechanism is unified and stable when establishing correlations between segments. The convolution blocks restrict modeling to short-range temporal neighborhoods and form stable local descriptors that serve as the fixed-scale foundation for higher-level aggregation.

B. CROSS-SEGMENT SPATIOTEMPORAL ATTENTION AND SEQUENCE MODELING

After preprocessing and convolution feature extraction, the obtained feature sequence needs to capture the temporal and spatial dependencies across segments. To this end, the sequence representation $Z^{(l)} = \{z_1^{(l)}, z_2^{(l)}, \ldots, z_T^{(l)}\}$ is input into the spatiotemporal attention module, and the query vector, key vector, and value vector are generated through the mapping matrix [21,22], which are in the form of:

$$Q = Z^{(l)}W_Q, \quad K = Z^{(l)}W_K, \quad V = Z^{(l)}W_V \tag{4}$$

In Formula (4), $W_Q, W_K, W_V \in R^{d \times d_a}$ are the linear transformation matrices of query, key, and value, respectively, and d_a represents the dimension of the attention space. The spatiotemporal correlation is achieved by calculating the similarity matrix, which is defined as:

$$A = Softmax\left(\frac{QK^\top}{\sqrt{d_a}} + M\right) \tag{5}$$

In Formula (5), $A \in R^{T \times T}$ represents the weighted relationship between time segments, and M is a cross-segment mask matrix that enforces segment-level constraints on attention propagation. The mask matrix assigns zero to valid frame pairs and negative infinity to invalid frame pairs so that suppressed entries do not contribute to the normalized attention weights. The value of each entry in M is determined by segment membership, where frame pairs within the same segment or temporally adjacent segments remain unmasked, while frame pairs from non-adjacent segments are blocked. All diagonal elements of M are set to zero to preserve self-attention at each temporal position. Keyframe positions arise from the attention response distribution during the query–key interaction, and the selection emerges through parameter optimization rather than predefined rules. Through this matrix, the corresponding keyframe and adjacent segment features receive higher weights, ensuring that local continuity of actions is maintained under global temporal aggregation [23,24].

After the cross-segment correlation is calculated, the distribution of attention weights reflects the relative temporal contribution of frames within each segment. To quantify the effect of attention aggregation, Table 1 reports the statistical properties of frame-level attention weights aggregated at the interval level before and after attention weighting. The attention weight associated with each frame is obtained by applying softmax normalization to the similarity scores along the temporal dimension within each input segment, ensuring that the weights are non-negative and that the total weight inside each segment equals one, while no normalization is imposed across the entire sequence. As a result, attention statistics are computed independently for each temporal interval without coupling their magnitude across different segments. The mean values reported in Table 1 represent interval-level average attention intensity obtained by averaging frame weights inside each time interval rather than across the entire sequence.

TABLE 1. Segment-level attention weight statistics in different time periods

Time Interval (frames)	Mean before weighting	Variance before weighting	Mean after weighting	Variance after weighting
0–20	0.18	0.052	0.24	0.019
21–40	0.21	0.061	0.27	0.022
41–60	0.19	0.048	0.26	0.017
61–80	0.22	0.055	0.29	0.020
81–100	0.20	0.059	0.25	0.018

From the results in Table 1, the interval-level attention weights exhibit higher mean values and lower variance after weighting, indicating that attention redistributes temporal importance within segments rather than uniformly across the entire sequence. The concentration of attention on a subset of frames reflects the emergence of dominant temporal responses during aggregation, leading to a more stable and discriminative representation in the temporal domain.

The cross-fragment correlation visualization heat map in Figure 3 illustrates the average distribution of spatiotemporal attention weights across all samples in the validation set, highlighting consistent weight patterns along the temporal dimension.

Figure 3 shows the aggregated attention weight distribution obtained by averaging attention matrices across all validation samples after temporal alignment. Peak weights align with action transitions and direction changes, showing that keyframe selection is fully learned end-to-end from attention scores. The horizontal and vertical axes correspond to the time step index; the highlight of the diagonal area reflects the continuity between adjacent frames; the high-weight block in the distal area reveals the capture of long-range dependencies; the aggregation weight of the keyframe position reflects the model's sensitivity to the turning point of the action. This visualization reflects stable attention patterns consistently observed across samples rather than isolated behavior from individual sequences.

The aggregated spatiotemporal feature is expressed as:

$$H = AV \tag{6}$$

In Formula (6), $H \in R^{T\times d_a}$ represents the representation sequence containing cross-fragment dependencies. The attention projections adjust the contribution of each frame in training so that high-response frames form the set of keyframes used in temporal aggregation. This representation retains the convolution feature in the spatial position and strengthens the overall semantics of the keyframe and continuous action in the temporal dimension. To capture long-range dependencies, the spatiotemporal attention output is input into a bidirectional long short-term memory network. Assuming the forward and backward hidden states are $\overrightarrow{h}_t$ and $\overleftarrow{h}_t$, respectively, the bidirectional representation is defined as:

$$h_t = \overrightarrow{h}_t \oplus \overleftarrow{h}_t \tag{7}$$

In Formula (7), $\oplus$ represents the vector concatenation operation. The recurrent encoder preserves ordered temporal transitions by propagating framewise states in both directions while focusing on local continuity and transition dynamics. This recurrent encoding does not impose explicit global structural constraints across distant segments.

In the bidirectional long short-term memory unit, the nonlinear mapping of the input gate, forget gate, and output gate prevents the gradient attenuation of the feature in the long-range transmission. Assuming the input is h_{t-1}, and the current feature H_t, the state update process is:

$$\begin{aligned} f_t &= \sigma(W_f H_t + U_f h_{t-1} + b_f),\\ i_t &= \sigma(W_i H_t + U_i h_{t-1} + b_i),\\ o_t &= \sigma(W_o H_t + U_o h_{t-1} + b_o),\\ \tilde{c}_t &= \tanh(W_c H_t + U_c h_{t-1} + b_c),\\ c_t &= f_t \odot c_{t-1} + i_t \odot \tilde{c}_t,\ h_t = o_t \odot \tanh(c_t), \end{aligned} \tag{8}$$

In Formula (8), f_t, i_t, and o_t are the activation results of the forget gate, input gate, and output gate, respectively; c_t is the candidate memory unit; $\odot$ represents the Hadamard product. This structure models temporal continuity through ordered state propagation, while cross-segment relations degrade under noise during sequence unfolding.

C. ACTION CLASSIFICATION AND BOUNDARY SMOOTHING

After cross-segment spatiotemporal modeling and bidirectional temporal encoding, the obtained sequence representation is input to the fully connected layer to map the high-dimensional features to the action category space. Assuming input features are $\{h_1, h_2,....., h_T\}$, the mapping process in the classification layer is:

$$y_t = Softmax(W h_t + b) \tag{9}$$

In Formula (9), $W\in R^{C\times d_h}$, $b\in R^C$, C represent the number of action categories, and d_h represents the hidden layer dimension. The output sequence $\{y_1, y_2,...,y_T\}$ pro-vides a frame-level category distribution at each time step, where each y_t corresponds to the predicted action proba-bility of the t-th frame derived from long-term contextual modeling. Supervision is applied at the frame level, and each predicted distribution is directly matched with the ground truth label of the corresponding frame without segment-level pooling or temporal aggregation.

In continuous interactive actions, segment boundaries often produce mutations. If classification training relies solely on cross-entropy loss, it is easy to cause category jumps at the boundaries. To alleviate this problem, a boundary smoothing constraint is applied, and the difference in predicted distribution between adjacent frames is embedded as a regularization term in the objective function [25,26]. The boundary smoothing loss is defined as:

$$L_{smooth} = \frac{1}{T-1}\sum_{t=1}^{T-1} \|y_{t+1} - y_t\|_2^2 \tag{10}$$

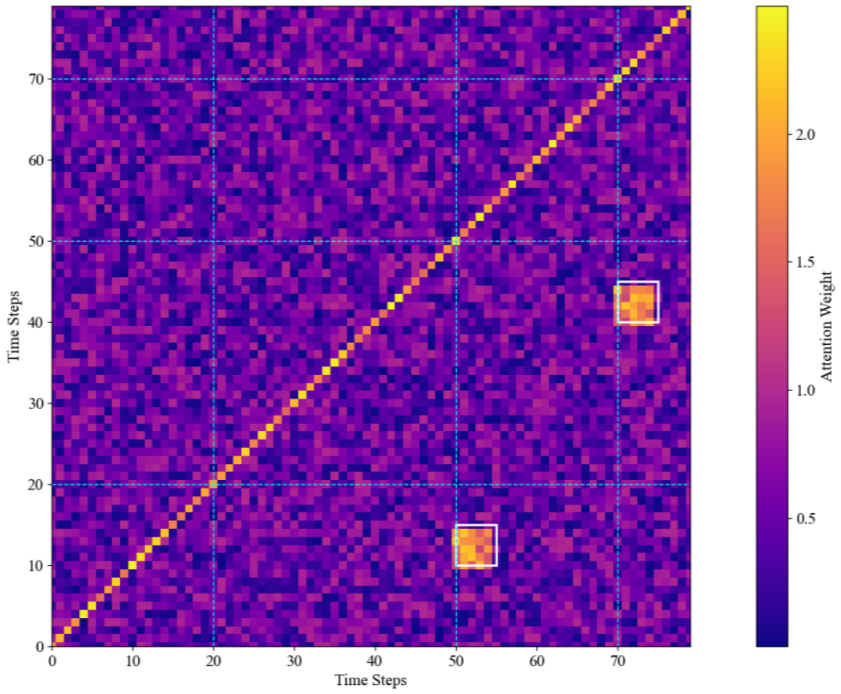

FIGURE 3. Average cross-fragment spatiotemporal attention weight distribution heat map over the validation set

This term measures the Euclidean distance of the category distribution between adjacent frames and constrains the continuity of the output in the time dimension, making the action boundary transition smooth and avoiding the fragmentation caused by category mutation. This constraint operates on adjacent-frame prediction distributions while keeping frame-level labels fixed, thereby regularizing boundary confidence without introducing temporal boundary offset. The classification loss is computed in a frame-wise manner over the entire sequence, enforcing temporal alignment between predicted frame labels and annotated frame-level ground truth. An action boundary is identified at a frame index where the ground truth label changes, and a predicted boundary is considered correct when it lies within a $\pm$ 5-frame tolerance window of the annotated boundary, based on which boundary precision, recall, and F1 score are computed. The frame- wise classification loss is defined using the cross-entropy formulation, which is given by:

$$L_{cls} = -\frac{1}{T}\sum_{t=1}^{T}\sum_{c=1}^{C}\delta_{c,t}\log y_{t,c} \tag{11}$$

In Formula (11), $\delta_{c,t}$ is the true category indicator function at time step t, taking the value of 1 when the frame belongs to category c and 0 otherwise; Through this objective function, the model enforces frame- wise classification accuracy while preserving temporal smoothness across adjacent frames.

D. BEHAVIOR PATTERN GRAPH MODELING AND JOINT OPTIMIZATION

After the action classification and boundary smoothing are completed, the obtained sequence labels and feature vectors are uniformly represented as a graph structure to reveal the global dependency between actions. The graph layer encodes global behavior patterns through structured dependencies that are not expressed through sequential propagation, forming a complementary modeling scale to the recurrent encoder. The graph treats each segment-level descriptor as a node and enforces relations between non-adjacent segments through learned edge weights so that global pattern constraints and non-sequential dependencies are represented explicitly rather than emerging indirectly from framewise recurrence. It is assumed that the node at time step t is represented as:

$$v_t = f(h_t, y_t) \tag{12}$$

In Formula (12), $h_t \in R^{d_h}$ denotes the temporal feature extracted from the sequence encoder, representing motion dynamics and contextual continuity, while $y \in R^C$ denotes the frame-level category confidence distribution produced under frame-wise supervision. Rather than serving as a final prediction outcome, y_t functions as a semantic conditioning signal that provides category-aware context for node construction. The function f(·) represents a learnable linear fusion mapping that concatenates ht and yt and projects the fused representation into the node embedding space through a fully connected transformation. This design preserves the dominance of feature-driven representation while injecting lightweight semantic guidance, avoiding reverse

dependence of structural modeling on predicted results. To ensure that graph topology is determined solely by intrinsic temporal feature similarity, edge construction is decoupled from category confidence information. The edge weight between nodes is determined by the spatiotemporal correlation matrix, which is defined as:

$$e_{ij} = \exp\left(-\frac{\|h_i - h_j\|_2^2}{\sigma^2}\right) \tag{13}$$

In Formula (13), e_{ij} represents the similarity between node i and node j measured in the temporal feature space, and σ controls the decay rate of similarity with respect to feature distance. To obtain a sparse adjacency structure, edge weights with similarity values below a fixed threshold are removed during graph construction, so that only dom-inant temporal relations are preserved. By computing edge weights solely from h_t, the graph topology reflects intrinsic temporal motion similarity, while node embeddings retain semantic enrichment from category confidence information, jointly ensuring structural stability and semantic coherence of continuous actions.

During the propagation process of the GNN, node features are iteratively updated between adjacent nodes. Topological aggregation preserves structural relations of the action sequence across long temporal spans, and the update rule is:

$$h_i^{(l+1)} = \sigma\left(\sum_{j \in N(i)} \alpha_{ij} W^{(l)} h_i^{(l)}\right) \tag{14}$$

In Formula (14), $h_i^{(l)}$ represents the feature of node i at the l-th layer; α_{ij} denotes the edge weight normalized by applying a softmax operation over all incoming edges connected to node i, such that the sum of α_{ij} over j equals one; $W^{(l)}$ is the trainable weight matrix at the l-th layer; $\sigma(\cdot)$ is the nonlinear activation function. In the loss function design, in addition to the action classification loss Lcls and the boundary smoothing loss Lsmooth, the graph consistency loss is also applied to constrain the difference between the predicted graph structure and the true action pattern. The adjacency matrix of the true pattern A* is constructed directly from the annotated action sequence of each sample by encoding the ground-truth temporal transitions between action segments, where an edge is assigned between two nodes if a transition occurs consecutively in the annotated sequence. Assuming that the adjacency matrix of the predicted graph is $\hat{A}$, and the adjacency matrix of the true pattern is A^*, then the loss is defined as:

$$L_{graph} = \left\|\hat{A} - A^*\right\|_F^2 \tag{15}$$

In Formula (15), $\|\cdot\|_F$ is the Frobenius norm. The matrix A^* is sample-specific and varies across different interaction sequences, reflecting the unique temporal ordering and dependency structure defined by the corresponding ground-truth annotations rather than a shared template or a global statistical prior. This constraint ensures that the dependency relationship between nodes is preserved in the predicted graph, so that the action sequence modeling is consistent with the behavior structure.

The overall objective function of the joint optimization is:

$$L = L_{cls} + \lambda_1 L_{smooth} + \lambda_2 L_{graph} \tag{16}$$

In Formula (16), λ_1 and λ_2 are the weight coefficients of the smoothing and graph consistency constraints, respectively, and are set to 0.4 and 0.2 based on empirical stability of training dynamics, where both constraints provide effective regularization while preserving the dominance of the frame-wise classification objective. Through this objective function, the model simultaneously optimizes classification accuracy, temporal smoothness, and structural consistency during training, ensuring the integrated realization of interactive action recognition and pattern modeling.

The full pipeline forms a hierarchical structure. Local spatiotemporal descriptors, attention-driven relevance weighting, frame-by-frame sequence continuity, and graph-level structure constraints operate at distinct modeling scopes under joint optimization.

IV. EXPERIMENTAL DESIGN AND IMPLEMENTATION

A. DATASET AND ENVIRONMENT CONFIGURATION

The VR interaction dataset used in the experiment is a self-collected dataset acquired through a tethered virtual reality system, covering three categories of user interaction behaviors: basic actions, interactive actions, and complex actions, corresponding to local body displacements, human–object interactions, and multimodal behaviors with long-term temporal dependencies. Data are collected using an HTC Vive Pro Eye head-mounted display, two Vive Controllers, and three Vive Trackers under a SteamVR runtime on a PC- based VR platform. The interaction system is implemented in Unity 2021 LTS, and synchronized six-degree- of-freedom pose data are accessed via the SteamVR SDK at a fixed sampling rate of 90 Hz. Each frame records three-dimensional position and orientation information from eleven tracking points, forming a temporally aligned multi-channel motion sequence. All sequences are annotated at the frame level following a predefined action taxonomy, where action boundaries are defined at semantic intent transitions and each frame is assigned to a single, mutually exclusive action category. Labeling is performed independently by two trained annotators, and consistency is verified using Cohen's Kappa coefficient, yielding an average agreement score of 0.87 after consensus correction. The dataset contains data from 24 participants, including 15 male and 9 female subjects aged 21 to 38 years. A subject-independent protocol is adopted for evaluation, ensuring that data from each participant appear exclusively in the training, validation, or test set. In terms of dataset statistics, the reported frame length reflects the total duration of each original interaction

sequence before segmentation, measured as the number of frames from sequence start to end. The average frame length and standard deviation in Table 2 therefore describe the temporal span of complete action samples rather than the fixed segment length used as network input. The results are shown in Table 2.

TABLE 2. Data size statistics for the three action categories

Action Category	Number of Samples	Average Frame Length	Standard Deviation of Frame Length
Basic Actions	520	45	6.2
Interactive Actions	480	72	8.5
Complex Actions	300	128	14.3

The results in Table 2 show significant differences in the number and duration of the three action types. Complex actions have significantly longer frame lengths than basic and interactive actions. The increased standard deviation also indicates a more uneven temporal span, posing a challenge for subsequent models in modeling long-term dependencies.

B. PARAMETER SETTING AND TRAINING STRATEGY

Before inputting the interactive action sequences into the network after unified frame rate and spatial normalization, model parameters must be set to ensure convergence stability and generalization. The three- dimensional convolution uses a fixed kernel size to maintain local consistency of spatiotemporal features. The convolution outputs undergo batch normalization and residual connections to maintain a stable numerical distribution. The combined architecture of cross-segment spatiotemporal attention and bidirectional long short-term memory networks relies on weight initialization and dynamic learning rate adjustments during training to ensure gradual convergence of long-term dependencies.

The training batch size and sequence length jointly influence video memory usage and timing capture range. While maintaining repeatability, the experiment fixes the batch size for all models. The number of training epochs is set within the post-convergence stable range, and updates are stopped when validation set performance no longer improves to avoid overfitting. To ensure controllability of the training process, Table 3 records the ranges of key training parameters, including convolution kernel size, batch size, initial learning rate, decay strategy, and maximum number of epochs.

TABLE 3. Key parameter values for model training

Parameter	Value Range	Final Value	Description
Convolution Kernel Size	$3\times3\times3$–$5\times5\times5$	$3\times3\times3$	Maintain local consistency
Batch Size	8–32	16	Balance memory and convergence
Initial Learning Rate	0.01–0.0005	0.001	Dynamic decay
Decay Strategy	0.9 per 10 epochs	0.9 per 10 epochs	Avoid oscillation
Maximum Epochs	50–200	120	Stable convergence range

Table 3 shows that the selected parameter values are all in the middle of the range, demonstrating a balance between convergence speed and stability. The fixed convolution kernel size of $3\times3\times3$ ensures stable feature extraction, while the learning rate and decay strategy ensure smooth numerical convergence during training.

C. VALIDATION AND TESTING PROCESS

Based on the subject-independent evaluation protocol, the experimental data is divided into training, validation, and test sets at a ratio of 70%, 15%, and 15%, where the split is performed at the participant level rather than the sequence level. All interaction sequences from a given participant are assigned exclusively to a single subset, ensuring that no subject appears in more than one split and preventing identity-specific motion patterns from leaking across training and evaluation stages. This ensures a consistent basis for comparing basic actions, interactive actions, and complex actions of different models during the analysis phase. The training set is used for parameter updates; the validation set monitors convergence trends and adjusts the weight coefficient of the joint loss term; the test set is used exclusively for performance evaluation.

V. RESULTS

A. ACTION RECOGNITION ACCURACY

In the action recognition accuracy analysis section, three representative comparison models are constructed to verify the effectiveness of the proposed method in interaction scenarios of varying complexity. Model 1, the Inflated 3D ConvNet (I3D), relies on 3D convolutions to capture spatial and local temporal features of video frame sequences, demonstrating robust performance in short-duration action recognition. Model 2, a combination of Convolutional 3D (C3D) and Bidirectional Long Short-Term Memory network (BiLSTM), extracts spatial features based on 3D convolutions and then extends temporal modeling through a recurrent network. This model demonstrates superiority over I3D in continuous action recognition. Model 3, the proposed joint modeling framework of 3D convolutions combined with cross-segment spatiotemporal attention and GNNs, emphasizes the representation of cross-segment dependencies and structured action relationships. The experimental action data is divided into three categories: basic actions focus on local body displacements or short-term movements; interactive actions represent the operational relationship between humans and virtual objects; complex actions encompass long-term dependencies and multimodal combinations. Figure 4 shows a comparison of the recognition accuracy of the three types of actions using the three models. All three models are evaluated using the same input representations, preprocessing procedures, and subject-independent data splits described in Section “Experimental Design and Implementation”. Training strategies and optimization settings are kept consistent across models to ensure that performance differences are attributed to modeling capability rather than experimental configuration. The reported average accuracy in each action group is computed using

macro-averaging over action categories. Per-class accuracy is calculated independently and averaged with equal weight, mitigating the influence of category frequency differences, which is more pronounced in complex actions.

The comparison results show that the proposed method maintains its advantage in all three action recognition tasks. In the basic action group, the proposed model achieves an average accuracy of 91.8%, exceeding I3D's 87.9% and C3D+BiLSTM's 88.6%. This difference is primarily due to the cross-segment attention mechanism reducing feature redundancy between similar short-duration actions. The interactive action group achieves an average accuracy of 90.4%, a significant improvement compared to I3D's 83.9% and C3D+BiLSTM's 85.3% (compared to basic actions). This is due to the more stable representation of the spatiotemporal dependencies between actions and virtual objects in the graph structure modeling. The largest improvement occurs in the complex action group, where the proposed model achieves an average accuracy of 87.2%, significantly outperforming I3D's 80.1% and C3D+BiLSTM's 81.6%.

B. ACTION BOUNDARY DETECTION

In the VR action boundary detection experiment, four typical models are applied for performance comparison. Model 1, a 3D-CNN with 3D convolution as its core, relies solely on local spatiotemporal convolutions to extract features. Model 2, a BiLSTM, focuses on capturing sequential temporal dependencies while ignoring spatial context. Model 3, a GNN, models inter-action dependencies using a graph structure but does not impose boundary constraints on action continuity. Model 4 is the cross-fragment spatiotemporal attention and graph structure modeling method proposed in this paper, which jointly optimizes convolutional features, attention mechanism and boundary smoothing loss. The experimental setup is divided into three categories based on action complexity: basic action group, interactive action group, and complex combination action group. This describes the model's boundary detection performance in scenarios of varying difficulty. Figure 5 shows the F1 score comparison of each model under the three action groups. Boundary detection is evaluated at the frame level, where a predicted boundary is regarded as correct if it lies within a $\pm$ 5-frame window of the annotated boundary. The F1 score is computed using macro-averaging across action categories.

In the basic action group, the models perform similarly. The average F1 score for 3D-CNN is around 80%, while those for BiLSTM and GNN are around 83%–84%. The proposed model achieves nearly 89%. This advantage stems primarily from the stability of short-term boundaries maintained by the convolutional and attention modules. The difference in the interactive action group increases significantly. The 3D-CNN maintains an overall F1 score between 69.60% and 73.40%, while the average F1 scores for BiLSTM and GNN increase to 74.70% and 76.90%, respectively. The proposed model generally exceeds 84%. This difference reflects the contribution of the boundary smoothing loss in scenes with ambiguous action start and end. The comparison is most striking in the complex combination action group. The 3D-CNN performance consistently remains in the 59.80% to 65.30% range, while the average F1 scores of the BiLSTM and GNN models increase to approximately 68.14% to 71.86%. The proposed model maintains a stable F1 score exceeding 79%, with the largest improvement compared to the GNN for continuous gestures. This demonstrates the full potential of the joint graph structure consistency constraint in scenarios with dense long-range dependencies and action interweaving.

C. TEMPORAL DEPENDENCY MODELING

In this experiment, to address the problem of maintaining temporal dependencies of continuous interactive actions in VR scenes, an action sequence of 100 frames is constructed and divided into four typical categories. Walking actions last 20 frames; grasping actions last 10 frames; waving actions last 25 frames; rotation actions last 45 frames. These include short-duration actions and long-duration actions, making it easier to test the performance of the model over different time spans. The comparison method selects three representative models. Model 1 is the BiLSTM, and its structure can process time series data but often exhibits fragmentation under long-range dependencies. Model 2 is a method that combines three- dimensional convolution with a Transformer encoder; it has the ability to model global dependencies across time segments, but has limited boundary continuity. Model 3 is a proposed method based on three- dimensional convolution, cross-segment spatiotemporal attention, and bidirectional long short-term memory networks; it uses GNNs to model behavioral patterns, which is used to unify action recognition and structure preservation. Figure 6 shows the differences between different methods in modeling long-range temporal dependencies.

Figure 6 shows that Model 1 exhibits segmentation and insertion errors for long actions such as walking and waving. This is because the model relies on a recursive structure to gradually update the sequence. Long- range dependency information decays during repeated state transfer, and short bursts of action interfere with memory units, resulting in over-segmentation of continuous actions. Model 2 is more stable in terms of boundary fit, relying on a self-attention mechanism to capture global dependencies. However, it exhibits sequence fragmentation for long-duration actions such as rotation. This is due to excessive averaging of global features, which results in insufficient constraints on the internal consistency of the action and causes boundary positions to shift across long segments.

D. STRUCTURAL CONSISTENCY OF BEHAVIOR PATTERN GRAPHS

In this experiment, the structural consistency of user interaction behavior sequences in VR is analyzed by comparing multiple GNN-based models. Method 1, a GNN-only method, uses a basic GNN and propagates features

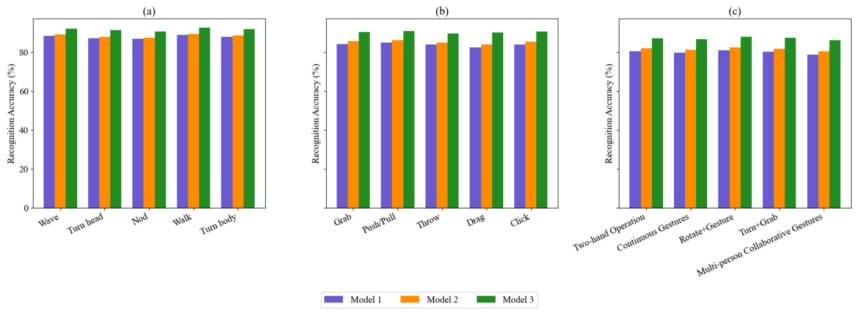

FIGURE 4. Comparison of VR user interaction action recognition accuracy; (a) Basic action group; (b) Interactive action group; (c) Complex action group

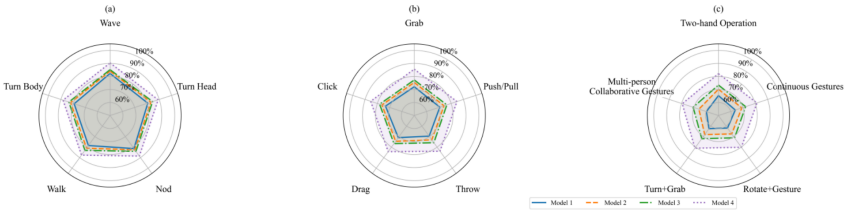

FIGURE 5. Comparison of F1 scores for continuous action boundary detection under three action groups; (a) Basic action group; (b) Interactive action group; (c) Complex combination action group

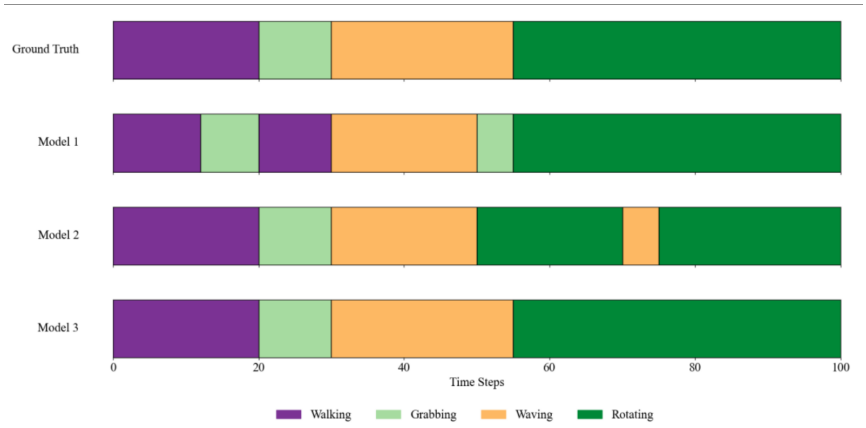

FIGURE 6. Alignment of action sequence prediction results

solely through spatiotemporal dependencies between nodes, without applying any additional constraints. Method 2, GNN + Boundary Smoothing Loss, adds a boundary smoothing constraint to the action sequence based on the GNN-only method to enforce local consistency of continuous actions. Method 3, GNN + Graph Consistency Loss, optimizes graph structural consistency to improve the accuracy of the overall behavior pattern graph. Method 4, GNN + a combination of the two losses, optimizes global graph consistency while maintaining local boundary constraints, achieving coordinated optimization of action sequences and structural patterns. The experiments are divided into two groups. The first group focuses on consistency dynamics during training and analyzes convergence behavior by comparing the percentage change in Graph Edit Distance-sim (GED-sim) over 100 training epochs. The second group examines the impact of action sequence length on model structural consistency and evaluates the robustness in complex scenarios by analyzing consistency performance for sequences of varying lengths. GED-sim is defined as a normalized structural similarity measure derived from graph edit distance. For a predicted graph G_p and the corresponding ground-truth graph G_t, GED-sim is computed as

$$GED\text{-}sim = 1 - \frac{GED(G_p, G_t)}{\max\left(|G_p|, |G_t|\right)} \tag{17}$$

where $GED(\cdot)$ denotes the minimum edit cost required to transform one graph into the other, and $|G|$ denotes the number of nodes. This normalization constrains GED-sim to the range [0, 1], enabling direct comparison across behavior graphs of different sequence lengths and scales. Figure 7 presents the comparative trends of the consistency indicators

of the four methods in terms of the number of training rounds and the length of action sequences.

For the GNN-only model, the GED-sim starts from a relatively low value and rapidly increases to 72.0% during the early stages of training, reflecting the limited capability of basic graph propagation in capturing complex node dependencies. The model with boundary smoothing loss exhibits a more pronounced improvement in the early training phase and reaches a final GED-sim of 79.3%, indicating that local boundary constraints effectively enhance the continuity of action features. The model incorporating graph consistency loss achieves a GED-sim of 91.1% in the later stages of training, demonstrating that global structural optimization substantially improves the matching accuracy of the behavior pattern graph. The joint loss model attains the highest final structural consistency of 95.2%, highlighting the advantage of collaborative optimization of local boundaries and global graph structure.

Results under varying sequence lengths show a clear decreasing trend of GED-sim as sequence length increases. Specifically, GED-sim decreases from 75.3% to 56.8% for the GNN-only model, while the joint loss model decreases from 88.0% to 70.0%. This indicates that combining local boundary constraints with global structural optimization effectively mitigates structural degradation in long and complex action sequences, preserving the integrity of continuous actions and the consistency of behavior pattern graphs.

E. EFFECT OF MULTI-CLIP KEYFRAME WEIGHTING

In this experiment's visualization analysis, high-dimensional interaction features are first reduced using principal component analysis (PCA) to visually display their distribution characteristics in a two-dimensional space. The horizontal and vertical axes correspond to the first and second principal feature dimensions after dimensionality reduction, respectively. This coordinate system is used to represent the relative positions of different interactive actions in the low-dimensional embedding space. In VR interactive behavior research, the distribution differences of different action categories in the feature space directly affect action recognition and modeling. To examine the role of the cross-clip spatiotemporal attention mechanism in keyframe feature aggregation, five typical interactive actions (grab, rotate, click, drag, and release) are selected. The feature embedding results before and after weighting are compared for the same batch of action samples. Keyframes are represented by large dots, and ordinary frames are represented by small dots, which intuitively present the adjustment effect of the attention mechanism on the feature distribution of action sequences, as shown in Figure 8.

The distribution before weighting shows a mixture of keyframes and regular frames, with relatively loose class boundaries, indicating that features have not yet formed clear category clusters during the temporal modeling process. After weighting, keyframes gradually shrink to the center of the category; the class distribution widens; the boundaries become clearer.

F. MODEL REAL-TIME AND ROBUSTNESS EVALUATION

In VR interactive scenarios, recognition models must not only maintain high accuracy but also meet real-time requirements and demonstrate robustness in complex environments. To this end, three representative deep spatiotemporal modeling methods are selected for comparison. Model 1, I3D, relies on 3D convolution to directly extend the 2D convolution kernel structure. While stable for short-term action recognition, it has limited efficiency for long-term inference. Model 2, C3D, combines BiLSTM with convolution to extract spatial features and extends temporal dependencies through a recurrent network. This method offers advantages for continuous action modeling, but suffers from limitations in inference speed and robustness. Model 3, proposed in this paper, combines 3D convolution with cross-segment spatiotemporal attention and graph structure modeling, emphasizing the capture of long-range dependencies while maintaining the integrity of behavioral structure. Experiments are conducted across three dimensions: real-time inference frame rate, recognition accuracy under occlusion, and boundary detection stability under motion blur. All inference experiments are performed on a single NVIDIA RTX 3090 GPU with batch size set to 1. The reported frame rate reflects pure model forward inference time and excludes data preprocessing overhead. The results are summarized in Figure 9.

For the occlusion evaluation in Figure 9(b), occlusion is simulated by randomly masking a fixed percentage of each frame with continuous spatial blocks, where the occluded regions vary across frames. The occlusion ratio denotes the proportion of masked pixels.

For the motion blur evaluation in Figure 9(c), motion blur is applied using a linear blur kernel, and the blur intensity is controlled by the kernel length, with larger kernels corresponding to stronger blur effects.

The results in Figure 9 show that when the input length is 32 frames, the inference frame rate of the proposed method is 39.2 FPS, higher than the 33.5 FPS of C3D+BiLSTM and the 29.8 FPS of I3D. This value reflects pure model inference time on an RTX 3090. In practical VR training systems, the recognition module executes in parallel with rendering, maintaining end-to-end latency well within the requirements of tethered VR headsets. When the length is extended to 128 frames, the three methods drop to 28.9 FPS, 22.1 FPS, and 18.2 FPS, respectively, demonstrating that the proposed method maintains high real-time performance even with long sequences. When the occlusion ratio is 0, the recognition accuracy of the proposed method reaches 91.8%, higher than the 88.6% of C3D+BiLSTM and 87.9% of I3D. When the occlusion ratio increases to 40%, the proposed method maintains 79.8%, while the C3D+BiLSTM and I3D rates drop to 71.5% and 64.2%, respectively, demonstrating its greater robustness to occlusion interference. Under

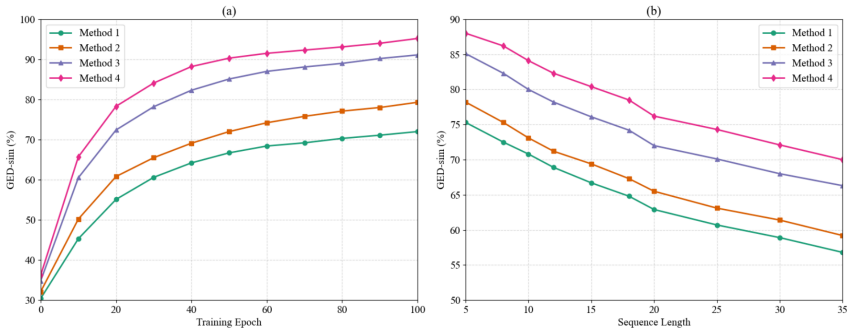

FIGURE 7. Structural consistency analysis of VR user interaction behavior sequence graphs; (a) Consistency index dynamics under varying training rounds; (b) Comparison of consistency indexes under varying action sequence lengths

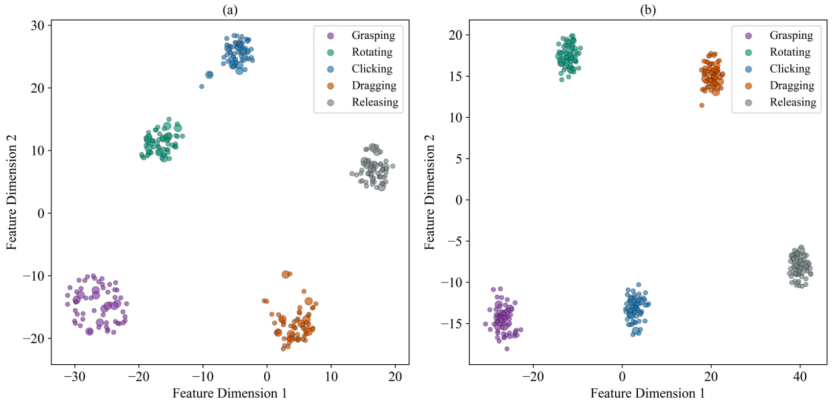

FIGURE 8. Comparison of interactive action feature distributions before and after keyframe weighting; (a) Feature distribution before weighting; (b) Feature distribution after weighting

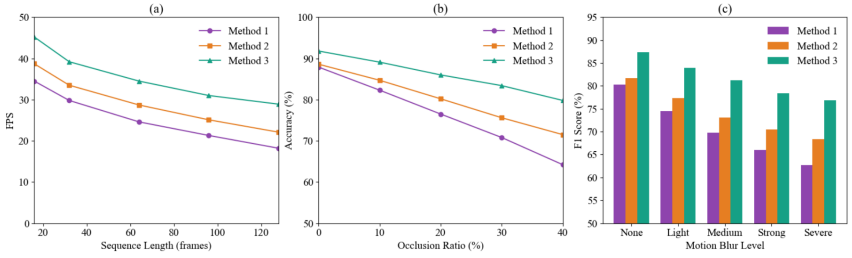

FIGURE 9. Model real-time and robustness evaluation; (a) Comparison of inference frame rate under different input sequence lengths; (b) Comparison of recognition accuracy under varying occlusion ratios; (c) Comparison of average F1 scores under varying action blur intensity

unblurred conditions, the average F1 score of the proposed method is 87.4%, surpassing the 81.7% of C3D+BiLSTM and 80.3% of I3D. As the blur intensity increases to severe levels, the average F1 score of the proposed method drops to 76.9%, while C3D+BiLSTM and I3D reach 68.4% and 62.7%, respectively.

G. MODULE CONTRIBUTION

The model consists of a 3D convolutional structure, a cross-segment spatiotemporal attention module, a bidirectional long short-term memory network, a behavioral structure modeling graph network, and boundary smoothing constraints. Each module undertakes independent modeling responsibilities in different dimensions of the action sequence. The 3D convolutional structure extracts local spatial and short-term dynamic features; spatiotemporal attention strengthens keyframe associations during cross-segment aggregation; the bidirectional long short-term memory network maintains sequential associations during sequence unfolding; the behavioral graph network maintains structural stability across cross-segment relationships; boundary smoothing constraints stabilize the output distribution within the continuity of the predicted sequence. To verify the role of each module in the complete framework, recognition experiments are conducted after removing modules one by one, maintaining a consistent verification process across the three action groups. The statistical results of action recognition accuracy are shown in Table 4 below.

TABLE 4. Ablation Comparison of Action Recognition Module

Model Setting	Basic Actions (%)	Interactive Actions (%)	Complex Actions (%)
Full Model	91.8	90.4	87.2
w/o At- tention	89.3	86.1	79.4
w/o GNN	90.6	85.7	81.3
w/o BiLSTM	88.4	82.6	74.8
w/o Boundary Smoothing	90.2	84.1	77.6

Note: w/o indicates the model settings created after removing the corresponding module.

The results show that the complete model maintains stable performance across the three action categories, reaching 87.2% for complex action groups. After removing cross-segment attention, the performance of complex action groups drops to 79.4%, indicating a weakening of the cross-segment aggregation ability and a loosening of the association between keyframes and continuous actions. After removing graph structure, the performance of complex action groups remains at 81.3%, but the lack of cross-segment topology causes a shift in the structural relationships within long chains during the propagation phase. After removing the bidirectional long short-term memory network, all three action categories show significant declines, with the performance of complex action groups dropping to 74.8%, indicating that the lack of temporal encoding paths makes the dependency arrangement within long sequences unstable. After removing boundary smoothing constraints, the performance of complex action groups drops to 77.6%, with the classification distribution at the boundaries fluctuating in continuous segments, thus affecting the recognition consistency of long segments. Overall, the results show that the model achieves multi-dimensional complementarity in temporal modeling, structural relationship preservation, and boundary continuity, thereby maintaining stable output in complex interactive actions.

VI. CONCLUSION

This paper proposes a cross-segment spatiotemporal attention and interactive behavior pattern modeling method based on a three-dimensional convolutional neural network, with immediate applications in enhancing the fidelity and precision of training simulations for complex tasks in the high-tech industry.

The sequential module maintains frame-wise updates, while the graph topology preserves cross-segment structure, forming complementary relations in long sequences. By capturing long-range dependencies through convolutional feature extraction, spatiotemporal attention aggregation, and a bidirectional long short-term memory network, this method combines boundary smoothing constraints with GNN structural consistency optimization to achieve holistic recognition and pattern modeling of user interactive behaviors in VR. Experimental results demonstrate that this method significantly improves recognition accuracy over traditional methods. The average accuracy for complex action groups reaches 87.2%, exceeding the 80.1% of I3D and the 81.6% of C3D+BiLSTM, demonstrating better preservation of long-range dependencies and action continuity. In continuous action boundary detection, the F1 score for complex combination action groups exceeds 79%, significantly outperforming the approximately 70% achieved by BiLSTM and GNN methods. In graph structural consistency modeling, GED-sim, optimized with a joint loss, achieves 95.2%, an improvement of over 23 percentage points over single constraint methods. This demonstrates that the coordinated optimization of local boundaries and global structure effectively enhances pattern recovery capabilities. The overall framework of this method ensures the accuracy of action classification while avoiding fragmentation and semantic loss of interactive behavior sequences. This allows for the full modeling of the continuity and structured relationships of user interaction processes in virtual reality environments, supporting intelligent analysis and human-computer collaboration in immersive training, education, and rehabilitation scenarios, and offering specialized support for skills training and quality control.

AUTHOR CONTRIBUTIONS

Huiling Wu designed, collected and analyzed the data, and drafted the manuscript. Huiling Wu conducted the study, critically revised the manuscript for important intellectual content, and gave final approval of the version to be published. Huiling Wu participated fully in the work, take public responsibility for appropriate portions of the content, and agreed to be accountable for all aspects of the work in ensuring that questions related to the accuracy or integrity of any part of the work are appropriately investigated and resolved.

CONFLICTS OF INTEREST

The author declares no conflict of interest.

FUNDING

This research received no external funding.

ACKNOWLEDGEMENTS

Not applicable.

REFERENCES

[1] C. Cheng and H. Xu, "A 3D motion image recognition model based on 3D CNN-GRU model and attention mechanism," Image and Vision Computing, vol. 146, no. 1, pp. 104991-105003, 2024, doi: 10.1016/j.imavis.2024.104991.

[2] M. Liu, W. Li, B. He, C. Wang, and L. Qu, "Human Action Recognition Based on 3D Convolution and Multi-Attention Transformer," Applied Sciences, vol. 15, no. 5, pp. 2695, 2025, doi: 10.3390/app15052695.

[3] M. Li, F. Pan, Y. Zhang, T. Chen, and H. Du, "Integrating deep learning model and virtual reality technology for motion prediction in emergencies," Safety Science, vol. 183, no. 1, pp. 106721-106733, 2025, doi: 10.1016/j.ssci.2024.106721.
[4] A. Kamali Mohammadzadeh, E. Alinezhad, and S. Masoud, "Neural-Network-Driven Intention Recognition for Enhanced Human–Robot Interaction: A Virtual-Reality-Driven Approach," Machines, vol. 13, no. 5, pp. 414-434, 2025, doi: 10.3390/machines13050414.
[5] H. Dawood, M. Nawaz, T. Nazir, A. Javed, A. K. J. Saudagar, and H. S. AlSagri, "ARNet: Integrating Spatial and Temporal Deep Learning for Robust Action Recognition in Videos," Computer Modeling in Engineering & Sciences (CMES), vol. 144, no. 1, pp. 1-14, 2025, doi: 10.32604/cmes.2025.066415.
[6] H. Ullah and A. Munir, "Human action representation learning using an attention-driven residual 3dcnn network," Algorithms, vol. 16, no. 8, pp. 369-391, 2023, doi: 10.3390/a16080369.
[7] X. Huang and Z. Cai, "A review of video action recognition based on 3D convolution," Computers and Electrical Engineering, vol. 108, no. 1, pp. 108713-108725, 2023, doi: 10.1016/j.compeleceng.2023.108713.
[8] P. Raimbaud, R. Lou, F. Danglade, P. Figueroa, J. T. Hernandez, and F. Merienne, "A task-centred methodology to evaluate the design of virtual reality user interactions: A case study on hazard identification," Buildings, vol. 11, no. 7, pp. 277-299, 2021, doi: 10.3390/buildings11070277.
[9] B. Gan, C. Zhang, Y. Chen, and Y. C. Chen, "Research on role modeling and behavior control of virtual reality animation interactive system in Internet of Things," Journal of Real-Time Image Processing, vol. 18, no. 4, pp. 1069-1083, 2021, doi: 10.1007/s11554-020-01046-y.
[10] M. Zong, R. Wang, Z. Chen, M. Wang, X. Wang, and J. Potgieter, "Multi-cue based 3D residual network for action recognition," Neural Computing and Applications, vol. 33, no. 10, pp. 5167-5181, 2021, doi: 10.1007/s00521-020-05313-8.
[11] H. Zhang, Z. Hu, D. Yu, L. Guan, X. Liu, and C. Ma, "Multipath attention and adaptive gating network for video action recognition," Neural Processing Letters, vol. 56, no. 2, pp. 124-144, 2024, doi: 10.1007/s11063-024-11591-3.
[12] A. Sánchez-Caballero, D. Fuentes-Jiménez, and C. Losada-Gutiérrez, "Real-time human action recognition using raw depth video-based recurrent neural networks," Multimedia Tools and Applications, vol. 82, no. 11, pp. 16213-16235, 2023, doi: 10.1007/s11042-022-14075-5.
[13] H. Li, X. Li, L. Su, D. Jin, J. Huang, and D. Huang, "Deep spatio-temporal adaptive 3d convolutional neural networks for traffic flow prediction," ACM Transactions on Intelligent Systems and Technology (TIST), vol. 13, no. 2, pp. 1-21, 2022, doi: 10.1145/3510829.
[14] G. L. Zhang, "Design of virtual reality augmented reality mobile platform and game user behavior monitoring using deep learning," International Journal of Electrical Engineering & Education, vol. 60, no. 2_suppl, pp. 205-221, 2023, doi: 10.1177/0020720920931079.
[15] C. Linse, H. Alshazly, and T. Martinetz, "A walk in the black-box: 3D visualization of large neural networks in virtual reality," Neural Computing and Applications, vol. 34, no. 23, pp. 21237-21252, 2022, doi: 10.1007/s00521-022-07608-4.
[16] H. Li, "Convolutional Neural Network-Based Virtual Reality Real-Time Interactive System Design for Unity3D," Computational Intelligence and Neuroscience, vol. 2022, no. 1, pp. 2530836-2530848, 2022, doi: 10.1155/2022/2530836.
[17] N. Tasnim and J. H. Baek, "Deep learning-based human action recognition with key-frames sampling using ranking methods," Applied Sciences, vol. 12, no. 9, pp. 4165-4183, 2022, doi: 10.3390/app12094165.
[18] S. Guan, H. Lu, L. Zhu, and G. Fang, "AFE-CNN: 3D skeleton-based action recognition with action feature enhancement," Neurocomputing, vol. 514, no. 1, pp. 256-267, 2022, doi: 10.1016/j.neucom.2022.10.016.
[19] Y. Ming, F. Feng, C. Li, and J. H. Xue, "3D-TDC: A 3D temporal dilation convolution framework for video action recognition," Neurocomputing, vol. 450, no. 1, pp. 362-371, 2021, doi: 10.1016/j.neucom.2021.03.120.
[20] E. M. Saoudi, J. Jaafari, and S. J. Andaloussi, "Advancing human action recognition: A hybrid approach using attention-based LSTM and 3D CNN," Scientific African, vol. 21, no. 1, pp. 01796-01816, 2023, doi: 10.1016/j.sciaf.2023.e01796.
[21] B. Chen, H. Tang, Z. Zhang, G. Tong, and B. Li, "Video-based action recognition using spurious-3D residual attention networks," IET Image Processing, vol. 16, no. 11, pp. 3097-3111, 2022, doi: 10.1049/ipr2.12541.
[22] M. Toshpulatov, W. Lee, S. Lee, H. Yoon, and U. Kang, "DDC3N: Doppler-driven convolutional 3D network for human action recognition," IEEE Access, vol. 12, no. 1, pp. 93546-93567, 2024, doi: 10.1109/ACCESS.2024.3422428.
[23] S. Pehlivan and J. Laaksonen, "Temporal teacher with masked transformers for semi-supervised action proposal generation," Machine Vision and Applications, vol. 35, no. 3, pp. 36-51, 2024, doi: 10.1007/s00138-024-01521-7.
[24] Y. You, L. Zhang, P. Tao, S. Liu, and L. Chen, "Spatiotemporal transformer neural network for time-series forecasting," Entropy, vol. 24, no. 11, pp. 1651-1669, 2022, doi: 10.3390/e24111651.
[25] N. Aziere and S. Todorovic, "Multistage temporal convolution transformer for action segmentation," Image and Vision Computing, vol. 128, pp. 104567-104577, 2022, doi: 10.1016/j.imavis.2022.104567.
[26] M. Li, X. Wang, H. Wang, and M. Yang, "LTGS-Net: Local Temporal and Global Spatial Network for Weakly Supervised Video Anomaly Detection," Sensors, vol. 25, no. 16, pp. 4884-4901, 2025, doi: 10.3390/s25164884.