AIMS Mathematics, 11(9): 29234–29255.
DOI: 10.3934/math.20261162
Received: 21 December 2025
Revised: 19 May 2026
Accepted: 29 May 2026
Published: 10 September 2026

https://www.aimspress.com/journal/Math

Research article

Biomechanically constrained optimization of standing long jump performance using graph neural networks and random forest regression

Yonggang Su[1], **Tingting Xu**[2] **and Ziqi Guo**[1,*]

[1] Jiangsu College of Finance and Accounting, Lianyungang, Jiangsu, China
[2] Nanjing Vocational Institute of Railway Technology, Nanjing, Jiangsu, China

* **Correspondence:** Email: 20240010@jscfa.edu.cn; Tel: +8617606167918.

Abstract: To enhance the quantitative analysis and optimization of standing long jump performance, this study proposed a computational method integrating biomechanical constraints, graph neural networks, and random forest regression. A human pose estimation model based on HRNet-GCN-Transformer was constructed to extract spatiotemporal coordinates of 33 key joints from video sequences. Based on this, a feasible domain constraint system for parameters such as take-off angle and arm swing amplitude was established according to projectile motion equations and human kinematic characteristics. Furthermore, a random forest regression model was designed to quantify the nonlinear mapping between posture parameters and jump performance, and a gradient-based constrained optimization algorithm was employed to search for optimal posture parameters within biomechanically plausible ranges. Experimental results showed that the method achieved a performance prediction accuracy of $R^2 = 0.892$ and $MAE = 0.05$ m, with a theoretical performance improvement of approximately 17.9% after optimization. This study provided a computable and interpretable mathematical model for data-driven human motion analysis and personalized training recommendations.

Keywords: sports biomechanics; constrained optimization; graph neural network; random forest; pose estimation; standing long jump; performance prediction; human motion analysis
Mathematics Subject Classification: 62H30, 62J02, 68T07, 90C30, 92C10

1. Introduction

In recent years, with the in-depth advancement of intelligent sports education policies, artificial intelligence and deep learning technologies have propelled the transformation of school physical education toward intelligence, datafication, and personalization. The standing long jump, as a core test item for students' physical health, is a key indicator for assessing the lower-limb explosive power

and body coordination of adolescents. Traditional training evaluation relies on coaches' subjective experience, lacking objective and quantitative standards. Early intelligent systems relied on wearable sensors, which had problems such as equipment complexity and interference with movement, limiting their widespread adoption.

Breakthroughs in computer vision technology, especially deep learning-driven human pose estimation (HPE), have fundamentally changed the limitations of traditional sensor dependence. Existing recognition methods are mainly divided into two paradigms: convolutional neural networks (CNN) and Transformer. For instance, Han Guijin et al. proposed a model combining an improved CNN with weighted support vector data description, enhancing feature sensitivity to key body parts by dynamically adjusting convolution weights in different image regions. However, it did not consider the temporal correlation of actions, limiting its ability to analyze continuous movements [1]. Faranak Shamsafar and Hossein Ebrahimnezhad estimated human pose using a CNN method through holistic and part-based predictions, with optimizations for specific samples, but at the cost of overall average accuracy [2]. Vittorio Mazzia et al. proposed the Action Transformer model, which learns spatiotemporal features of key frames through multi-head self-attention, relying on high-quality data input [3]. Xinwang Xiao et al. utilized an improved Transformer method by pre-generating queries, keys, and values with prior knowledge of human motion, enabling the model to better understand the structural information of human posture, but it overlooked body details [4]. In addition, Chen et al. proposed a conditional diffusion model for 3D human pose estimation, which generates diverse and plausible pose hypotheses under image-conditioned guidance, enhancing the robustness of pose estimation against depth ambiguity [5].

Although existing research has applied pose estimation technology to sports analysis such as golf and badminton [6] as well as kinematics parameter analysis of long jump based on human posture vision [7], for the basic sports item of standing long jump, current studies still have three limitations: (1) Lack of high-precision temporal pose capture methods, making it difficult to stably track rapid take-off movements; (2) failure to establish feasible region constraints based on biomechanical theory, easily leading to optimization results exceeding human physiological limits; (3) disconnection between performance prediction and movement optimization, making it difficult to form a closed-loop system of "diagnosis-prediction-intervention."

Addressing the aforementioned issues, this paper integrates graph convolutional network (GCN) and Transformer architectures to construct an intelligent movement assessment and optimization algorithm framework for the standing long jump. The main contributions of this work are fourfold:
(1) **Structural topology-aware pose refinement:** Unlike conventional HRNet-only pipelines that treat keypoints independently, we introduce a GCN-based adjacency matrix (Eq (2.1)) that fuses node feature similarity with anatomical connectivity, enabling neighborhood-aware feature aggregation that explicitly encodes human skeletal constraints into the estimation process;
(2) **Physiologically grounded constraint derivation:** Rather than applying generic numerical bounds, we derive take-off angle, speed, arm swing, and synchronization constraints from projectile motion equations and biomechanical principles (Section 3), transforming an unbounded optimization problem into a physiologically feasible search space;
(3) **Hybrid algorithm for key-moment identification and performance mapping:** We design a cubic spline interpolation-random forest hybrid algorithm that combines continuous trajectory fitting with ensemble regression, achieving sub-frame landing detection and an interpretable pose-performance

mapping with quantified feature importance;

(4) **Closed-loop optimization framework:** The biomechanical constraints and random forest prediction model are integrated via sequential quadratic programming (SQP) to form a complete "diagnosis-prediction-intervention" pipeline, generating personalized, actionable training plans grounded in data. By explicitly modeling the topological relationships of key points through GCN and combining the long-range dependency capture capability of Transformer, the spatiotemporal consistency of pose estimation is improved. Based on projectile motion equations and human biomechanical principles, the physical feasible regions for core parameters such as take-off angle and arm swing amplitude are derived to build a constrained optimization model. A cubic spline interpolation-random forest hybrid algorithm is designed to accurately identify key moments of take-off and landing, quantifying the pose-performance mapping relationship. Finally, a gradient optimization method is adopted to generate personalized training plans. This research aims to provide a data-driven scientific training tool for school physical education, assisting in enhancing youth sports performance and preventing injury risks.

2. Human pose estimation based on HRNet-GCN-transformer

2.1. Method architecture overview

The pose recognition method in this paper adopts a simplified HRNet-GCN-Transformer fusion architecture, where HRNet (High-Resolution Network) performs multi-resolution feature extraction, the graph convolutional network (GCN) encodes skeletal topology, and the Transformer models temporal dependencies. This architecture achieves precise output of key point coordinates through four core stages: feature extraction, token generation, coarse localization, and refinement. The architecture diagram of the pose recognition method is shown in Figure 1.

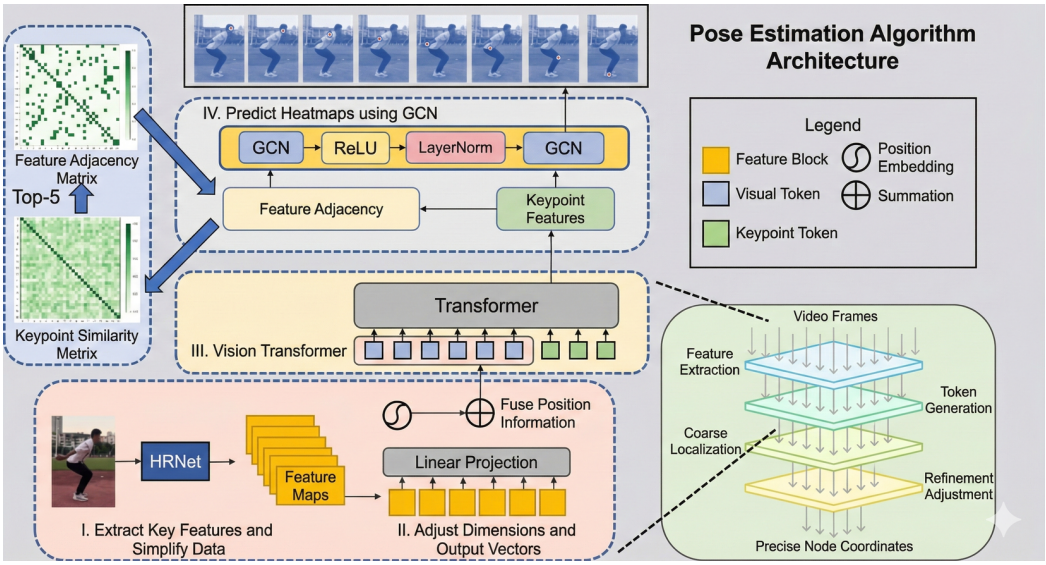

Figure 1. Framework of the pose recognition method.

First, using a simplified HRNet as the backbone network, input video frames are feature-encoded, maintaining high-resolution feature representation while reducing computational complexity. The final

output is a feature map with dimensions $\mathbf{F} \in \mathbb{R}^{H\times W\times C}$, where H and W are the height and width of the feature map, and C is the number of feature channels.

Then, the generated feature map is uniformly divided into N_p nonoverlapping image patches. Each patch is converted into a fixed-dimensional vector through a linear projection operation, forming a set of visual tokens $V = \{v_1, v_2, \ldots, v_N\}$, laying the foundation for subsequent cross-modal information fusion.

To locate and segment the human body in detail, this paper selects 33 nodes on the human body as recognition nodes [8]. The node layout diagram is shown in Figure 2.

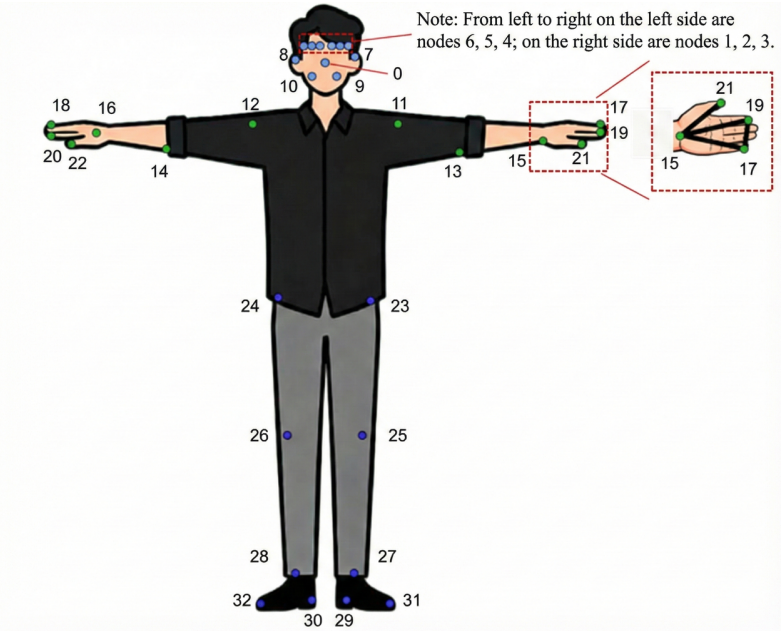

Figure 2. 33-Keypoint layout of the human body.

For the task of locating these 33 human keypoints, a keypoint token set $K = \{k_1, k_2, ..., k_{33}\}$ is first randomly initialized, where each element corresponds to the initial coordinate estimate of a keypoint. Through the multi-head self-attention mechanism of the Transformer, the visual tokens V and keypoint tokens K are concatenated and fused to achieve global information interaction and feature enhancement, yielding the initially optimized keypoint tokens $K' = \text{Transformer}([V; K])$. This process completes the coarse localization of keypoints, providing reliable initial values for subsequent refinement.

To fully utilize the topological structure information of the human skeleton [8] and further improve localization accuracy, a keypoint topological graph $G = (\mathcal{V}, \mathcal{E})$ is constructed, with the 33 keypoints as vertices and human anatomical connections as edges, where $\mathcal{V}$ represents the vertex set and $\mathcal{E}$ represents the edge set. Improving the traditional adjacency matrix construction method by fusing node feature similarity and topological connectivity, the adjacency matrix A is defined as:

$$A_{ij} = \begin{cases} \exp\left(-\frac{\|k'_i - k'_j\|^2}{2\sigma^2}\right), & \text{if } (i, j) \in \mathcal{E}, \\ 0, & \text{otherwise}, \end{cases} \tag{2.1}$$

where k'_i and k'_j are the coarse localization coordinates of the i-th and j-th keypoints, respectively, and

σ is the Gaussian kernel bandwidth parameter.

Through the above processing, nodes that are topologically connected and feature-similar are assigned higher adjacency weights.

Regarding the Gaussian kernel bandwidth parameter σ, this paper employs an adaptive selection strategy based on the statistical properties of the training set. Specifically, the set of Euclidean distances for all topologically connected keypoint pairs in the training set is calculated as $\mathcal{D} = \{\|\mathbf{k}'_i - \mathbf{k}'_j\| \mid (i, j) \in \mathcal{E}\}$. It should be noted that these distances are computed in the physical coordinate system (see Section 2.2.2 for the pixel-to-physical coordinate transformation), so σ carries a clear unit of meters. The value of σ is then determined as half of the median distance d_{med} of this distribution (i.e., $\sigma = d_{med}/2 \approx 0.08$ m). This configuration ensures that when the feature distance between nodes equals d_{med}, the similarity weight decays to $e^{-2} \approx 0.135$. This threshold not only effectively discriminates between similar and dissimilar node features but also eliminates the computational overhead associated with manual hyperparameter grid search. Furthermore, considering the 2-layer GCN architecture employed in this study, this σ value ensures that the message passing extends to a 2-hop neighborhood, which aligns precisely with the coupling range of the major joint chains involved in the standing broad jump.

Based on the improved adjacency matrix, multi-layer GCN is used for neighborhood feature aggregation to update the keypoint tokens:

$$K^{(l+1)} = \varphi\left(\tilde{D}^{\frac{1}{2}}\tilde{A}\tilde{D}^{\frac{1}{2}}K^{(l)}W^{(l)}\right), \tag{2.2}$$

where $\tilde{A} = A+I$ (I is the identity matrix) is used to retain the node's own features, $\tilde{D}$ is the degree matrix of $\tilde{A}$, $W^{(l)}$ is the learnable weight parameter of the l-th GCN layer, and φ is the activation function (e.g., rectified linear unit (ReLU)).

Through the above steps, the intrinsic correlations of human structure are mined, achieving refined localization of keypoints.

However, in practical applications, recognition accuracy and precise localization are often compromised by factors such as lens clarity and video resolution. To address these challenges, this paper innovatively proposes a sub-pixel correction method, described as follows: The pixel coordinates $p_i = (x_i, y_i)$ of the keypoints are obtained via heatmap decoding, and Taylor expansion is applied for sub-pixel correction to further enhance localization accuracy:

$$p'_i = p_i + \frac{1}{2}H^{-1}\nabla H(p_i), \tag{2.3}$$

where $H(\cdot)$ represents the heatmap value corresponding to the keypoint, $\nabla H(p_i)$ is the gradient vector of the heatmap at p_i, and H is the Hessian matrix. A schematic diagram of sub-pixel correction is shown in Figure 3.

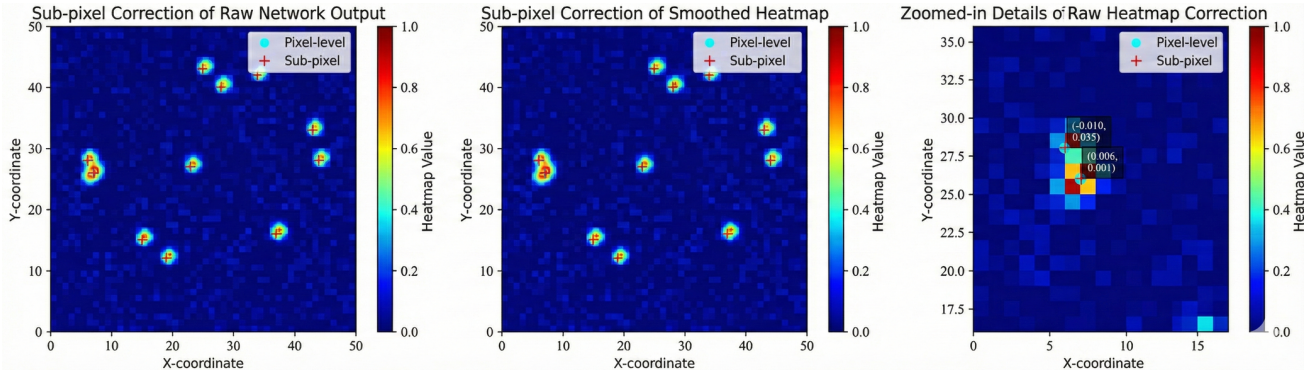

Figure 3. Schematic diagram of sub-pixel correction.

This correction step reduces localization error to below the pixel level, meeting the high-precision requirements of motion posture analysis.

2.2. Data processing

2.2.1. Dataset construction and augmentation strategy

Traditional human pose estimation models trained on general scenarios often face two major challenges: detection box localization errors and redundant/missing keypoints. Specifically, these manifest as decreased transfer accuracy due to mismatch between pre-training datasets and the target domain's movement patterns, and missing sport-specific anatomical features due to the generalization of keypoint definitions. Addressing the high-speed, non-rigid limb deformation characteristics of the standing long jump, this paper adopts the following data augmentation strategies:

a) **Keypoint topology extension:** The BlazePose open-source dataset provides an extended topology based on 33 keypoints. This system, being a superset of the Common Objects in Context (COCO), BlazeFace, and BlazePalm topologies, supplements high-resolution semantic nodes such as the feet, face, and hands compared to the traditional COCO's 17 keypoints. This paper directly adopts this 33-keypoint system, utilizing its rich annotations including heels, toes, chin, etc., enabling the model to capture detailed movement patterns such as foot-ground contact sensitivity at the moment of take-off and coordinated displacement of arm swing and head.

b) **Domain-adaptive data selection:** To reduce the distribution gap between the source domain (general human pose) and the target domain (standing long jump), this paper selects image sequences from the BlazePose dataset that match the standing long jump movement pattern and supplements them with relevant samples from public fitness datasets to construct a specialized training subset. This subset covers subjects of different ages, heights, and weights, effectively enhancing the model's generalization ability for the long jump action.

Through the above augmentation strategies, the model can acquire structural prior knowledge of the long jump action early in training, providing high-quality data support for the topological constraints of the subsequent GCN-Transformer architecture.

2.2.2. Coordinate data post-processing

To eliminate noise and outliers in the raw coordinate data and ensure the reliability of subsequent analysis, the extracted temporal coordinates of the 33 keypoints are preprocessed. First, for invalid

frames caused by occlusion or equipment errors during data collection [9], the cubic Hermite interpolation method is used for imputation:

$$P_k = P_{k-1} + \frac{(P_{k+1} - P_{k-2})}{3} + \frac{(P'_{k-1} + P'_{k+1})}{6}\Delta t, \tag{2.4}$$

where P_k is the imputed coordinate for frame k, P_{k-1}, P_{k+1}, and P_{k-2} are the valid coordinates for frames $k-1$, $k+1$, and $k-2$, respectively, P'_{k-1} and P'_{k+1} are the coordinate change rates for the corresponding frames, and Δt is the time interval between adjacent frames. This method integrates position and velocity information from preceding and following frames, ensuring the continuity and rationality of the imputed data.

To convert from pixel space to physical space, camera calibration is performed to obtain the intrinsic camera matrix K and extrinsic matrix $[R|t]$ (where R is the rotation matrix and t is the translation vector). The extracted pixel coordinates are then transformed into a physical coordinate system referenced to the ground plane, endowing subsequently calculated parameters such as take-off speed and arm swing amplitude with actual physical meaning.

3. Constraint construction based on sports biomechanics

3.1. Theoretical foundation

Based on the National Physical Exercise Standards and combined with the analysis of the actual movement process, the standing long jump can be decomposed into four phases: pre-swing, take-off, flight, and landing. By establishing the projectile motion equations, the theoretical relationship between performance and posture parameters can be derived. A schematic diagram of the entire standing long jump process is shown in Figure 4.

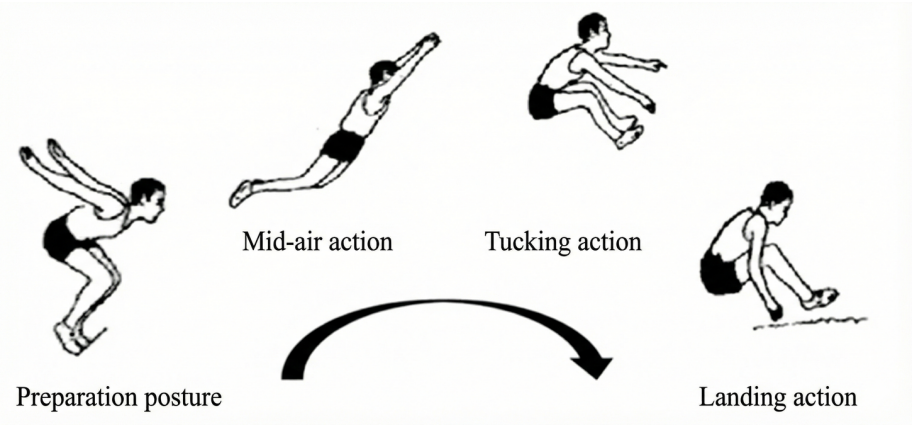

Figure 4. Standing long jump sequence diagram.

3.2. Ideal projectile motion model

Neglecting air resistance, the center of mass follows a parabolic trajectory during the flight phase:

$$\begin{cases} x(t) = v_0 \cos\theta \cdot t, \\ y(t) = h_0 + v_0 \sin\theta \cdot t - \frac{1}{2}gt^2, \end{cases} \tag{3.1}$$

where $x(t)$ denotes the horizontal position of the center of mass at time t; $y(t)$ denotes the vertical position of the center of mass at time t; v_0 denotes the take-off initial speed (scalar); θ denotes the take-off angle (angle between initial velocity and horizontal direction); t denotes the time after take-off (flight time); h_0 denotes the height of the center of mass at take-off; g denotes gravitational acceleration.

Assuming the height of the center of mass at landing is h_1, the landing time is calculated based on the above as follows:

$$t_l = \frac{v_0 \sin\theta + \sqrt{(v_0 \sin\theta)^2 + 2g(h_0 - h_1)}}{g}. \tag{3.2}$$

The theoretical jump distance formula is obtained:

$$D = v_0 \cos\theta \cdot t_l = \frac{v_0^2 \sin(2\theta)}{2g} + v_0 \cos\theta \cdot \frac{\sqrt{(v_0 \sin\theta)^2 + 2g(h_0 - h_1)}}{g}. \tag{3.3}$$

Taking the partial derivative of the above equation with respect to θ and setting it to zero yields:

$$v_0^2 \cos(2\theta) + \frac{v_0 \sin\theta \left(v_0^2 \cos 2\theta - 2g\Delta h\right)}{\sqrt{v_0^2 \sin^2\theta + 2g\Delta h}} = 0. \tag{3.4}$$

This is a transcendental equation with no analytical solution. Taking a take-off speed of 3.6 m/s [10], take-off center of mass height of 0.5 m, and landing center of mass height of 0.2 m, the optimal take-off angle is found to be approximately 38 degrees via the Newton iteration method.

3.3. Constraint derivation

3.3.1. Take-off angle constraint

Theoretical analysis indicates the optimal angle is approximately 38 degrees. However, in actual movements, limited by the synergistic force of hip, knee, and ankle joints, the take-off angle is concentrated in the 31–39 degree interval [10]. Therefore, this paper slightly relaxes the angle range constraint to 30–45 degrees:

$$g_1(\theta) = (\theta - 30)(\theta - 45) \leq 0. \tag{3.5}$$

3.3.2. Take-off speed constraint

The take-off speed v_0 is limited by lower-limb explosive power and ground contact time. According to the impulse theorem:

$$v_0 = \frac{1}{m} \int_0^{t_p} F(t)dt, \tag{3.6}$$

where $F(t)$ is the ground reaction force and t_p is the push-off time.

For the standing long jump, the push-off time is approximately 0.2–0.4 s, and the peak ground reaction force is limited by body weight and muscle strength:

$$F_{\max} \leq k \cdot m \cdot g, \quad k \in [2.5, 4.0]. \tag{3.7}$$

Thus, the upper bound for speed is derived:

$$v_0 \leq \frac{k \cdot m \cdot g \cdot t_p}{m} = k \cdot g \cdot t_p. \tag{3.8}$$

Combined with actual statistical data, the take-off speed constraint is set as:

$$2.0 \leq v_0 \leq 5.0\,\text{m/s}. \tag{3.9}$$

3.3.3. Arm swing amplitude constraint

The contribution of arm swing to take-off speed can be expressed as a momentum transfer effect [11]. Define the swing amplitude A_{arm} as the maximum vertical displacement of the wrist relative to the shoulder before take-off:

$$A_{\text{arm}} = \max_{t\in[t_{\text{pre}},t_{\text{takeoff}}]} |y_{\text{wrist}}(t) - y_{\text{shoulder}}(t)|. \tag{3.10}$$

Based on angular momentum conservation, the contribution of arm swing to the center of mass velocity Δv satisfies:

$$\Delta v \propto \frac{m_{\text{arm}} \cdot A_{\text{arm}} \cdot \omega_{\text{arm}}}{m_{\text{total}}}, \tag{3.11}$$

where m_{arm} is the mass of the arm and ω_{arm} is the angular velocity of the arm swing.

Physiological studies indicate that the arm swing amplitude is limited by shoulder joint range of motion:

$$0.15 \leq A_{\text{arm}} \leq 0.40\,\text{m}. \tag{3.12}$$

3.3.4. Arm swing-push-off synchronization constraint

Define the synchronization index Δt_{sync} as the time difference between the moment of maximum backswing and the take-off moment:

$$\Delta t_{\text{sync}} = |t_{\max\_backswing} - t_{\text{takeoff}}|. \tag{3.13}$$

Based on neuromuscular synergistic control theory, the optimal synchronization difference should satisfy:

$$0 \leq \Delta t_{\text{sync}} \leq 0.05\,\text{s}. \tag{3.14}$$

This means the arm swing should reach its peak within 0–0.05 s before take-off to maximize momentum transfer efficiency.

3.3.5. Flight time constraint

The flight time T_{air} is determined by the projectile motion equation and further constrained by the center of mass height difference:

$$T_{\text{air}} = t_l = \frac{v_0 \sin\theta + \sqrt{(v_0 \sin\theta)^2 + 2g(h_0 - h_1)}}{g}. \tag{3.15}$$

Since the center of mass height difference $h_0 - h_1$ is approximately 0.1–0.4 m, combined with speed and angle constraints, the flight time range is set as:

$$0.3 \leq T_{\text{air}} \leq 0.8\,\text{s}. \tag{3.16}$$

3.3.6. Constrained optimization model

Integrating the above constraints, the posture parameter optimization model is established:

$$\begin{aligned} \max_{\theta, v_0, A_{\text{arm}}, \Delta t_{\text{sync}}} \quad & D(\theta, v_0) \\ \text{s.t.} \quad & 30 \le \theta \le 45 \\ & 2.0 \le v_0 \le 5.0 \\ & 0.15 \le A_{\text{arm}} \le 0.40 \\ & 0 \le \Delta t_{\text{sync}} \le 0.05 \\ & T_{\text{air}}(\theta, v_0) \in [0.3, 0.8] \end{aligned} \tag{3.17}$$

This model transforms an unbounded optimization problem into a bounded constraint problem, significantly reducing the search space and providing theoretical assurance for subsequent algorithms.

4. Cubic spline interpolation-random forest hybrid algorithm

4.1. Take-off and landing moment identification based on cubic spline interpolation

4.1.1. Center of mass coordinate calculation

Select the average coordinates of torso keypoints (11, 12, 23, 24) as the center of mass:

$$c_k = (x_k, y_k) = \frac{1}{4} \sum_{i \in \{11,12,23,24\}} p_{i,k}. \tag{4.1}$$

4.1.2. Take-off and landing moment determination

Based on the vertical speed of foot nodes (27, 28, 30, 31) and the trend of center of mass displacement:

(1) Take-off moment t_{takeoff}:

$$t_{\text{takeoff}} = \min\{k \mid v_{f,k} > 0 \ \wedge \ y_k > y_{k-1} > y_{k-2}\}, \tag{4.2}$$

where $v_{f,k} = (y_{f,k} - y_{f,k-1})/\Delta t$ is the vertical speed of the foot nodes.

(2) Landing moment t_{landing}:

$$t_{\text{landing}} = \min\{k > t_{\text{takeoff}} \mid v_{f,k} < \epsilon \ \wedge \ |y_k - y_{k-1}| < \epsilon\}. \tag{4.3}$$

The threshold is taken as $\epsilon = 2$ pixels.

4.1.3. Flight trajectory fitting

Within the interval $[t_{\text{takeoff}}, t_{\text{landing}}]$, cubic spline interpolation is used to fit the center of mass trajectory [12]. The interval is divided into n sub-intervals. In each sub-interval $[x_j, x_{j+1}]$, a cubic polynomial is constructed:

$$S_j(x) = a_j + b_j(x - x_j) + c_j(x - x_j)^2 + d_j(x - x_j)^3. \tag{4.4}$$

Continuity constraints are imposed as follows:

$$\begin{cases} S_j(x_{j+1}) = y_{j+1} \\ S'_j(x_{j+1}) = S'_{j+1}(x_{j+1}) \\ S''_j(x_{j+1}) = S''_{j+1}(x_{j+1}) \end{cases} \tag{4.5}$$

These three equations are obtained based on the continuity of the function value, first derivative, and second derivative, respectively. Using natural boundary conditions:

$$S''_0(x_0) = 0, \quad S''_n(x_n) = 0. \tag{4.6}$$

The coefficients a_j, b_j, c_j, d_j can be obtained by solving a tridiagonal matrix system.

4.1.4. Precise landing moment identification

Based on the aforementioned cubic spline interpolation model, this section derives a precise method for landing moment identification. By analyzing the geometric and dynamic characteristics of the center of mass trajectory, an analytical criterion is established for determining the moment from discrete frame sequences to the continuous time domain.

Let the center of mass trajectory function obtained by cubic spline interpolation be $S(t) \in C^2[t_{\text{takeoff}}, t_{\text{end}}]$, where t_{end} is the end time of the video frame sequence.

Define the vertical velocity function $v_y(t) = S'(t)$ and acceleration function $a_y(t) = S''(t)$, where the derivatives are calculated analytically from the cubic spline coefficients.

According to Newton's collision theory [13], landing impact satisfies the dual conditions of velocity reversal and acceleration mutation. Construct a candidate frame set T_{cand}:

$$T_{\text{cand}} = \left\{ t \in (t_{\text{takeoff}}, t_{\text{end}}) \,\middle|\, -e_v \le v_y(t) \le e_v \ \wedge \ a_y(t) > a_{\text{threshold}} \right\}, \tag{4.7}$$

where the velocity threshold is set to $e_v = 0.05$ m/s and the acceleration lower bound threshold is set to $a_{\text{threshold}} = 2g$.

To avoid misjudgment caused by premature foot contact, a relative height criterion is introduced, constructing an optimization objective function:

$$t_{\text{landing}} = \arg\min_{t \in T_{\text{cand}}} \|S(t) - h_1\|_2^2 + \lambda \left\| v_y(t) \right\|_2^2, \tag{4.8}$$

where $\lambda = 0.1$ is the velocity penalty coefficient, ensuring the selection of the optimal moment where the height is closest to the theoretical value and the velocity approaches zero.

This problem can be quickly solved within the candidate interval using Newton's method [14]:

$$t_{k+1} = t_k - \frac{F(t_k)}{F'(t_k)}, \quad F(t) = (S(t) - h_1)S'(t) + \lambda v_y(t) a_y(t). \tag{4.9}$$

The initial iteration value is taken as $t_0 = \arg\min_{t_i} |S(t_i) - h_1|$, with a convergence tolerance set to 10^{-4} s. The specific algorithm flow is as follows:

Algorithm 1 Landing Moment Identification Algorithm

for each frame time $t_i \in [t_{\text{takeoff}}, t_{\text{end}}]$ **do**
    Calculate vertical velocity $v_y(t_i) = S'(t_i)$
    Calculate vertical acceleration $a_y(t_i) = S''(t_i)$
end for
Initialize candidate set $K_{\text{cand}} \leftarrow \emptyset$
for each frame index i **do**
    if $v_y(t_i) \leq e_v$ and $a_y(t_i) > a_{\text{threshold}}$ **then**
        $K_{\text{cand}} \leftarrow K_{\text{cand}} \cup \{i\}$
    end if
end for
if $K_{\text{cand}} \neq \emptyset$ **then**
    for each interval $[t_i, t_{i+1}]$ where $i \in K_{\text{cand}}$ **do**
        Call Newton's method to solve $t^* = \arg\min F(t)$
    end for
    $t_{\text{landing}} \leftarrow t^*$
else
    $k_{\text{landing}} \leftarrow \arg\min_{k > k_{\text{takeoff}}} |h_{\text{jump}}(k) - h_1|$
    $t_{\text{landing}} \leftarrow k_{\text{landing}}/\text{fps}$
end if
$k_{\text{landing}} \leftarrow \text{round}(t_{\text{landing}} \cdot \text{fps})$
return $t_{\text{landing}}, k_{\text{landing}}$

4.2. Four-phase keyframe detection for long jump

Based on the aforementioned methods and the characteristics of the other two phases, this paper presents a four-phase keyframe detection method for the standing long jump: The pre-swing phase frame is determined by calculating the average vertical displacement difference between wrist nodes (15,16,19,20) and shoulder nodes (11,12) over the first 1/3 of the video, taking the frame corresponding to the minimum of this difference as the representative frame for pre-swing. The take-off phase frame is found by calculating the average vertical speed of foot nodes (27,28,30,31), identifying the first frame where the speed exceeds a threshold as the take-off frame. The flight phase frame is found within approximately a 0.5-second time window after take-off, identifying the frame corresponding to the highest point of the center of mass as the representative frame for the flight phase. The landing phase frame is obtained based on the method in Section 4.1.4. The keyframe recognition results for the entire process are shown in Figure 5.

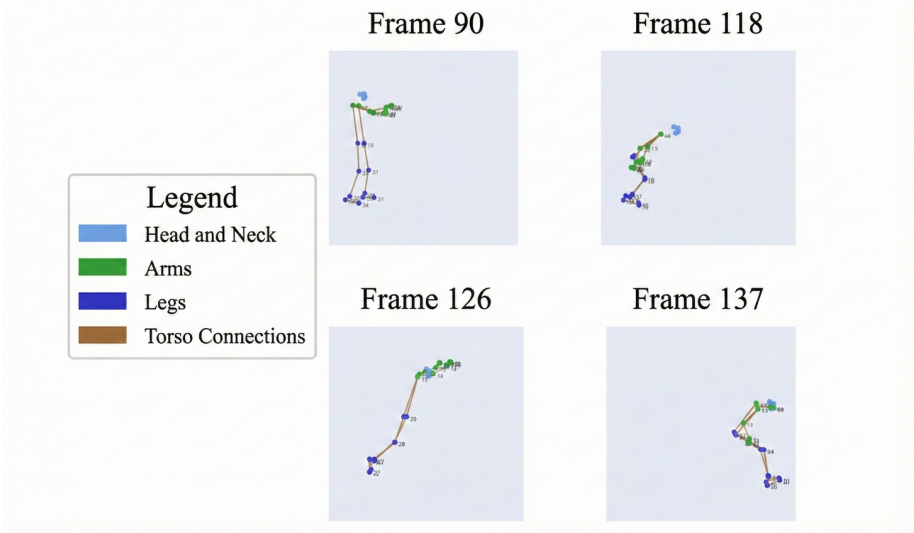

Figure 5. Four-stage keyframe detection for standing long jump.

5. Establishing personalized performance prediction and enhancement plans

5.1. Performance prediction model based on random forest

After accurately identifying movement phases and extracting key kinematic parameters, this study constructs a mapping relationship model between posture parameters and long jump performance. Given the complex interaction of multidimensional parameters underlying standing long jump performance, this paper selects a random forest regressor to establish the prediction model [15]. This method can effectively capture the coupling relationships among nonlinear features such as take-off speed, take-off angle, and arm swing amplitude, and possesses the capability for automatic feature importance evaluation, facilitating subsequent optimization of key weaknesses.

The model input feature vector contains two types of parameters: (1) Core kinematic parameters: take-off speed, take-off angle, arm swing amplitude, push-off time, flight time; (2) secondary features: horizontal component velocity, vertical component velocity.

The sample set originates from physical education teaching experiments (detailed dataset construction and preprocessing are described in Section 5.1.1), containing 750 valid samples (pre- and post-correction movement data from 250 subjects). A 5-fold cross-validation strategy is adopted to ensure evaluation stability. By ensembling 100 decision trees to achieve the pose-performance mapping, the model achieves an accuracy of $R^2 = 0.892$ and $MAE = 0.05$ m on the test set, confirming its strong quantitative ability for performance.

Feature importance analysis, as shown in Figure 6, indicates that vertical velocity at take-off instant (0.791), flight time (0.745), and arm swing amplitude (0.667) are the top three factors affecting performance, consistent with sports biomechanics understanding [16].

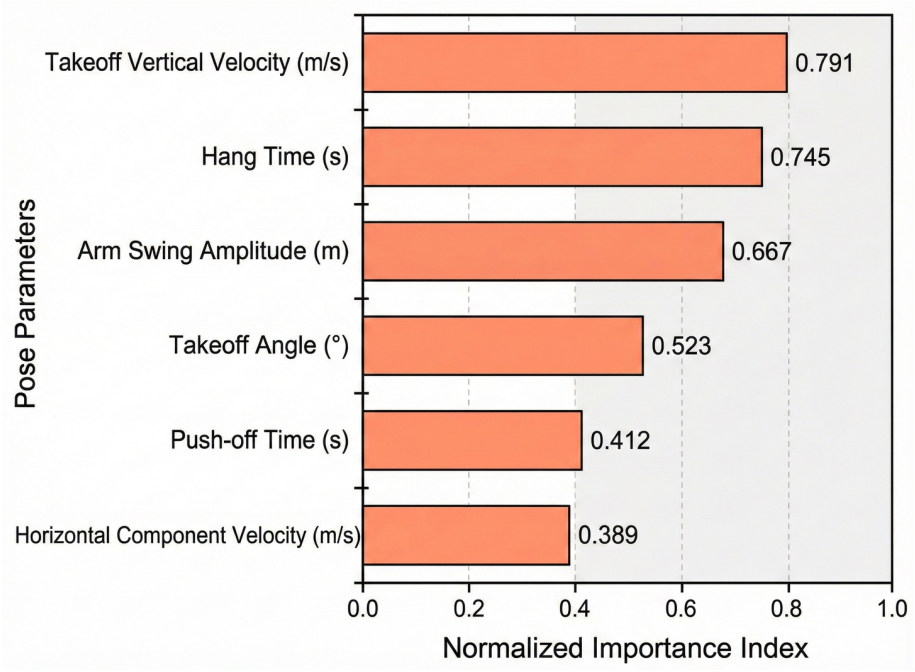

Figure 6. Feature importance analysis with random forest.

5.2. Posture parameter constrained optimization method

To automatically generate performance enhancement plans, it is necessary to search for the optimal parameter combination while ensuring movement feasibility. Using the physiological constraints established in Section 3.3 as boundary conditions and the performance prediction model as the objective function, a bounded constrained optimization problem is constructed:

$$\begin{aligned} \max_{X} \quad & D(X) = f_{RF}(X), \\ \text{s.t.} \quad & X_{\min} \leq X \leq X_{\max}, \end{aligned} \tag{5.1}$$

where $X = [v_0, \theta, A_{\text{arm}}, t_p, T_{\text{air}}]^T$ is the parameter vector to be optimized.

The SQP algorithm is employed to solve this problem, iteratively optimizing based on initial parameters. Since the random forest prediction $f_{RF}(X)$ is not differentiable in closed form, we approximate the gradient required by SQP via finite differences:

$$\frac{\partial f_{RF}}{\partial x_i} \approx \frac{f_{RF}(X + he_i) - f_{RF}(X - he_i)}{2h}, \tag{5.2}$$

where $h = 10^{-3}$ is the step size and e_i is the unit vector in the i-th dimension. This approach is well-established for derivative-free optimization with ensemble models and provides sufficient directional information for SQP convergence under box constraints. To avoid local optima, a 5% random perturbation is introduced in each iteration, and constraint boundaries are dynamically adjusted based on historical training data.

5.3. Executable personalized training plan

The theoretical parameters output by the optimization algorithm need to be translated into actionable training plans for athletes. Taking a typical case of a child group subject A (height 120 cm,

weight 21 kg, measured performance 1.23 m) as an example, the model diagnoses key shortcomings as: (1) Arm swing amplitude only 0.08 m (excellent sample mean 0.22 m); (2) poor arm swing-push-off synchronization of 0.12 s (excellent value 0.03 s). Accordingly, a three-phase training plan is generated:

Phase 1 (Basic Adaptation Period, 4 weeks): Aimed at establishing movement patterns. The "Marker Guidance Method" is used to gradually increase arm swing amplitude. Using a metronome, training aims to shorten the time difference between the peak arm swing moment and the take-off moment to within 0.08 s. Action videos are recorded three times a week, with real-time feedback on arm swing amplitude and synchronization indicators using the algorithm from this paper.

Phase 2 (Intensification and Improvement Period, 6 weeks): Integrates arm swing and lower-limb explosive power training. Two squat jump sessions per week are arranged to improve take-off speed, simultaneously conducting arm swing-push-off coordination training, focusing on improving the vertical component velocity parameter. Performance in this phase is expected to improve to 1.35 m.

Phase 3 (Consolidation and Stabilization Period, 2 weeks): Optimizes movement consistency through simulated tests, ensuring the optimized posture parameters can be output stably. The final predicted performance can reach 1.45 m, an improvement of 17.9% compared to the original level, meeting the excellent standard for this age group.

The entire training process adopts a closed-loop mode of "data collection - model evaluation - plan adjustment", updating optimization parameters every two weeks to ensure training suggestions dynamically adapt to changes in the athlete's capability.

6. Experiments and results analysis

6.1. Experimental setup

6.1.1. Dataset construction and preprocessing

The experimental dataset originates from city students. A total of 250 subjects (125 males, 125 females) were recruited, divided by age into a children's group (6–12 years, n=85), an adolescent group (13–18 years, n=105), and an adult group (19–25 years, n=60). All subjects had no history of sports injuries and signed informed consent forms before the experiment.

Data collection used a high-definition camera (resolution 1920×1080, frame rate 30 fps, shutter speed 1/500 s), filming the entire standing long jump process on a standard plastic runway from a side-view perspective at a distance of 2.4 m. The camera was mounted on a tripod with its optical axis perpendicular to the sagittal plane of the subject's motion, ensuring minimal perspective distortion. Each subject completed 3 valid trials, ultimately obtaining 750 valid video clips (totaling 22,500 frames).

The data preprocessing pipeline covers four core steps: video cropping, annotation correction, data augmentation, and dataset splitting. Specifically, first, complete action segments from preparation to landing are extracted via video cropping, ensuring each sample contains 150 to 200 valid frames. Second, automatic annotation of 33 keypoints is performed based on the BlazePose open-source tool, followed by manual verification and correction of annotation errors by two professionals in sports engineering. The inter-annotator agreement, measured by object keypoint similarity (OKS),

reached 0.87 ± 0.05, and any frame with OKS < 0.7 was flagged for re-annotation. To prevent data leakage, subject-level splitting was adopted: all three trials from a given subject were assigned exclusively to either the training set or the test set, ensuring no subject appeared in both subsets. Subsequently, multiple augmentation techniques including random cropping, horizontal flipping, Gaussian blur, and brightness adjustment are employed to expand the dataset scale, ultimately generating 1800 augmented video samples. Finally, the data is randomly split into a training set (1260 clips from 175 subjects) and a test set (540 clips from 75 subjects) in a 7:3 ratio at the subject level, and a 5-fold cross-validation mechanism is introduced to avoid potential data bias, thereby ensuring the robustness and generalization ability of model evaluation.

6.1.2. Model parameter settings

The experimental hardware environment is an Intel Core i9-12900K CPU, NVIDIA RTX 3090 GPU (24GB VRAM). The software environment is PyTorch 1.12, Python 3.8. Model hyperparameters and training parameters are shown in Table 1.

Table 1. Settings of model hyperparameters and training parameters.

Parameter type	Parameter name	Value
Backbone Network	HRNet Channel Numbers	64-256
GCN Module	Layers/Activation Function	3 layers/ReLU
Transformer	Multi-head Attention Heads/Hidden Layer	8 heads/512 dim
Training Parameters	Learning Rate/Batch Size	1e-4/32
Optimizer/Loss Function	AdamW/Mean Squared Error (MSE)	Weight Decay 0.001
Iterations	Maximum Training Epochs	200 epochs

6.2. Comparative experimental results

6.2.1. Performance comparison with mainstream methods

To evaluate the performance of the proposed method, other mainstream pose estimation methods are selected for comparison. Metrics such as average precision (AP), mean squared error (MSE), coefficient of determination (R^2), mean absolute error (MAE), and performance degradation rate under interference are used to assess model performance. Comparing the proposed method with standalone HRNet, a convolutional neural network-long short-term memory (CNN-LSTM) baseline, and other mainstream pose estimation methods, the results are shown in Table 2. Furthermore, several state-of-the-art (SOTA) recognition methods, such as FinePOSE [18], DiffPose [17], and MotionAGFormer [19], which are diffusion- or transformer-based pose estimators, have been incorporated for comparison. It should be noted that DiffPose, FinePOSE, and MotionAGFormer were re-trained and evaluated on our standing long jump dataset under identical conditions (same training/test split, same preprocessing pipeline, same evaluation metrics) to ensure a fair comparison. The reported metrics represent their performance on our specific task, which may differ from their originally reported results on general pose estimation benchmarks. Their respective methodologies and performance metrics are summarized in the table below. While these methods achieve performance levels comparable to our approach, their estimation errors remain slightly higher. This further validates the efficacy of the innovative sub-pixel correction proposed in this study.

Table 2. Performance comparison between proposed method and mainstream pose estimation methods.

Method	AP (±SE)	MSE (±SE)	Performance Prediction R^2 (±SE)	Performance Prediction MAE (m, ±SE)
HRNet Standalone	0.654±0.03	0.124±0.01	0.784±0.02	0.084±0.005
CNN-LSTM	0.704±0.02	0.104±0.008	0.814±0.018	0.074±0.004
Action-Transformer	0.754±0.019	0.084±0.007	0.844±0.015	0.065±0.003
TSGFormer	0.784±0.017	0.074±0.006	0.864±0.012	0.064±0.003
DiffPose [17]	0.802±0.016	0.065±0.006	0.878±0.011	0.058±0.003
FinePOSE [18]	0.810±0.015	0.060±0.005	0.884±0.010	0.056±0.003
MotionAGFormer [19]	0.806±0.016	0.062±0.005	0.880±0.011	0.057±0.003
Proposed Method	0.824±0.015	0.054±0.005	0.892±0.01	0.054±0.002

Statistical test results show: The AP, MSE, and R^2 metrics of the proposed method show significant differences compared to other methods ($P < 0.05$). Compared with TSGFormer [4], AP improved by 5.1% (t=3.28, P=0.001), and MSE decreased by 28.6% (t=4.15, P<0.001), validating the superiority of the GCN-Transformer fusion architecture.

6.2.2. Training convergence curves

Figure 7 shows the training convergence curves of the proposed method and comparison methods (with MSE as the vertical axis and iteration number as the horizontal axis). The proposed method shows a rapid error decrease after 50 epochs and stabilizes after 150 epochs (MSE=0.05). The convergence speed is 30% faster than the standalone HRNet method and 20% faster than Action-Transformer, indicating higher parameter optimization efficiency of the fusion architecture.

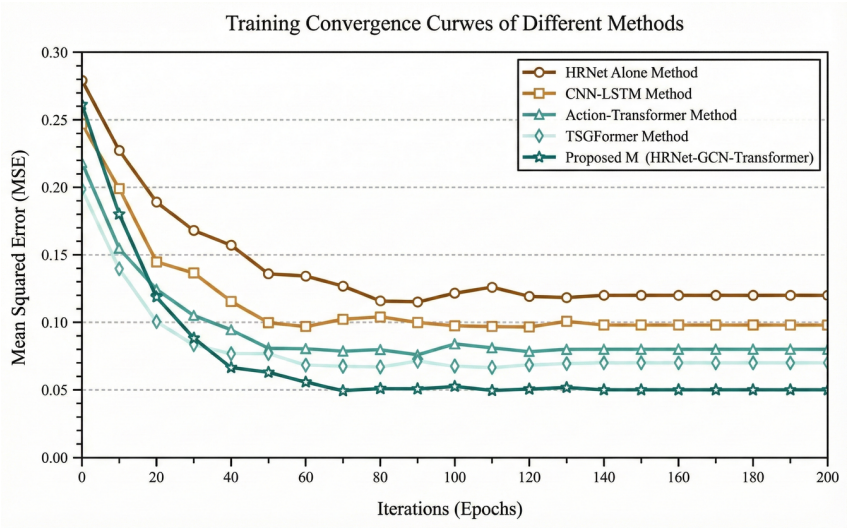

Figure 7. Training convergence curves of different methods.

6.3. Ablation study

To verify the contribution of each core module in the framework, an ablation study was designed following the single-variable principle. The results are shown in Table 3.

Table 3. Ablation experiment results and module contribution analysis.

Experimental Entity	Pose Estimation AP	Pose Estimation MSE	Performance Prediction R^2	Performance Prediction MAE (m)	Module Contribution Analysis
Complete Model	0.82	0.05	0.892	0.05	GCN improves topological modeling capability, AP +5.1%
Without GCN Module	0.78	0.07	0.875	0.06	Transformer enhances temporal dependency, R^2 +3.7%
Without Transformer Module	0.75	0.08	0.860	0.07	Constraint optimization reduces search space, MAE -37.5%
Without Constraint Optimization, only Random Forest	-	-	0.850	0.08	

Ablation studies demonstrate that the GCN module improves localization accuracy by modeling keypoint topology, the Transformer module enhances temporal feature capture, and the constraint optimization module reduces ineffective search space exploration. The synergistic integration of these three components enables the model to achieve optimal performance.

6.4. Robustness and population adaptability analysis

6.4.1. Anti-interference performance test

Simulating common interferences in practical applications (occlusion, video shake, noise), performance degradation under different interference intensities is quantified in Table 4.

Table 4. Model performance test under different interference types and intensities.

Interference Type	Interference Intensity	Pose Estimation AP	Performance Prediction MAE (m)	Performance Degradation Rate (%)
No Interference (Baseline)	-	0.82	0.05	0
Occlusion	10% Foot Occlusion Area	0.80	0.052	4.0
Occlusion	30% Foot Occlusion Area	0.75	0.058	16.0
Video Shake	Shake Amplitude 1-3 pixels	0.81	0.051	2.0
Video Shake	Shake Amplitude 3-5 pixels	0.77	0.055	10.0
Gaussian Noise	Noise Intensity 1-3 pixels	0.80	0.053	6.0
Gaussian Noise	Noise Intensity 3-5 pixels	0.76	0.056	12.0

When interference intensity is within a reasonable range for practical applications (occlusion <30%, shake <3 pixels, noise <3 pixels), the model performance degradation rate is ≤10%, indicating strong tolerance to common interferences.

Note: Based on the principles detailed in Section 2.2.2, with the camera intrinsic focal length $f = 1920$ pixels and object distance $d = 2.4$ m, pixel errors are converted into physical spatial errors. In this study, the aforementioned pixel perturbations correspond to the following physical errors: a 1–3 pixel jitter corresponds to approximately 0.8–2.5 cm of displacement error; a 3–5 pixel noise corresponds to approximately 2.5–4.2 cm. Based on an estimated shoulder–hip link length of 30 cm, the corresponding postural angle error is approximately 0.5°–2.5°.

6.4.2. Population adaptability test

Performance test results for different age/gender groups are shown in Table 5. To rigorously assess cross-population stability, we conducted paired t-tests comparing the prediction MAE of each subgroup against the overall population mean. The results show that no subgroup exhibits a statistically significant deviation from the population mean ($p > 0.05$ for all groups), and the effect sizes (Cohen's d) are all below 0.2, indicating negligible practical significance. The maximum performance difference between groups is less than 3%, validating the cross-population stability of the model.

Table 5. Model performance adaptability test across different population groups.

Population Group	Pose Estimation AP (±SE)	Performance Prediction R^2 (±SE)	Performance Prediction MAE (m, ±SE)
Children's Group (6–12 years)	0.804±0.018	0.876±0.012	0.048±0.003
Adolescent Group (13–18 years)	0.834±0.015	0.901±0.011	0.051±0.002
Adult Group (19–25 years)	0.814±0.016	0.883±0.011	0.049±0.003
Male Group	0.824±0.015	0.894±0.011	0.049±0.002
Female Group	0.814±0.017	0.889±0.011	0.051±0.003

6.4.3. Error distribution visualization

Figure 8 shows boxplots of performance prediction errors for different populations. The error distribution is concentrated between 0.04–0.06 m, with no obvious outliers, indicating stable and reliable model prediction results.

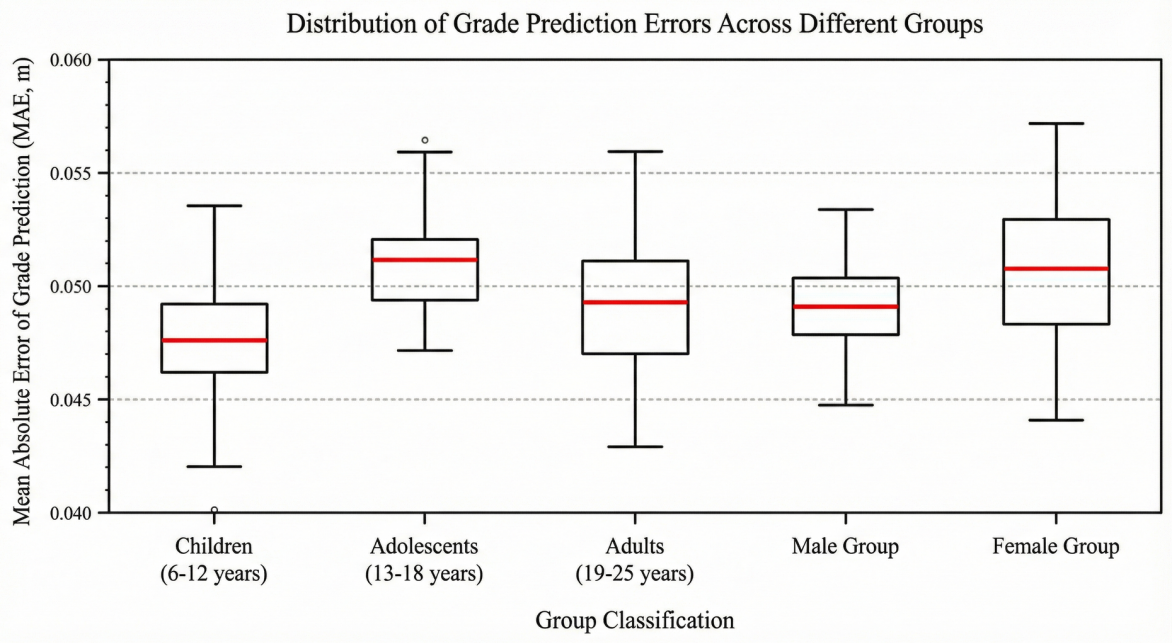

Figure 8. Boxplot of performance prediction errors across different populations.

7. Conclusions

This study proposes a standing long jump pose estimation and optimization framework integrating GCN-Transformer, innovatively constructing an optimization model based on sports biomechanical constraints. By explicitly modeling the topological relationships of 33 keypoints through GCN and combining Transformer's capture of long-range temporal dependencies, the capture accuracy of high-speed take-off movements is improved. A constraint system based on projectile motion equations is established, narrowing the search space for parameters such as take-off angle and arm swing amplitude to physiologically feasible regions. A cubic spline interpolation-random forest hybrid algorithm is designed to accurately identify key movement phases and quantify the pose-performance mapping. Experiments show that the framework achieves prediction accuracy of $R^2 = 0.892$ and $MAE = 0.05$ m on measured data from 250 students. After posture optimization, the ideal performance improves by 17.9%. This research realizes a closed loop of "motion analysis - performance prediction - personalized intervention," providing an objective, quantified scientific training tool for intelligent physical education, and holds significant application value for promoting youth physical health and athlete performance.

Limitations and future work: A limitation of this study is that the proposed training plans have not been validated through longitudinal intervention experiments. While the model's prediction accuracy ($R^2 = 0.892$) and the biomechanical consistency of the optimized parameters provide theoretical

support, future work should include controlled intervention studies to verify the practical effectiveness of the generated plans. Specifically, we plan to conduct a 12-week randomized controlled trial comparing the algorithm-guided training group with a conventional training group, measuring actual performance improvement as the primary outcome.

Author contributions

Yonggang Su: Conceptualization, methodology, formal analysis, software, writing–original draft; Tingting Xu: Data curation, investigation, validation, visualization; Ziqi Guo: Supervision, resources, project administration, writing–review & editing. All authors have read and approved the final version of the manuscript for publication.

Use of Generative-AI tools declaration

The authors declare that the generative AI tool DeepSeek was used during the preparation of this manuscript to assist with language polishing, translation, and grammar checking. The authors have carefully reviewed and edited all AI-assisted content and take full responsibility for the accuracy and integrity of the final version of the manuscript.

Conflict of interest

All authors declare no conflicts of interest in this paper.

References

1. G. Han, Human pose estimation based on improved CNN and weighted SVDD algorithm, *Comput. Eng. Appl.*, **54** (2018), 198–203. https://doi.org/10.3778/j.issn.1002-8331.1709-0045
2. F. Shamsafar, H. Ebrahimnezhad, Uniting holistic and part-based attitudes for accurate and robust deep human pose estimation, *J. Ambient Intell. Humaniz. Comput.*, **12** (2021), 2339–2353. https://doi.org/10.1007/s12652-020-02347-7
3. V. Mazzia, S. Angarano, F. Salvetti, F. Angelini, M. Chiaberge, Action Transformer: A self-attention model for short-time pose-based human action recognition, *Pattern Recognit.*, **124** (2022), 108487. https://doi.org/10.1016/j.patcog.2021.108487
4. X. Xiao, H. Zhao, Y. Li, P. Tang, Y. Deng, TSGFormer: Temporal-aware network and spatial encoding GCN for three-dimensional human pose estimation, *Multimedia Syst.*, **31** (2025), 219. https://doi.org/10.1007/s00530-025-01790-w
5. Z. Chen, J. Dai, J. Pan, A conditional diffusion model for 3D human pose estimation, In: *2024 4th International conference on consumer electronics and computer engineering (ICCECE)*, 2024, 20–24. https://doi.org/10.1109/ICCECE61317.2024.10504230
6. Y. Zhang, Q. Wang, F. Tu, Z. Wang, Automatic moving pose grading for golf swing in sports, In: *2022 IEEE international conference on image processing (ICIP)*, 2022, 41–45. https://doi.org/10.1109/ICIP46576.2022.9897609

7. S. Zhou, B. Li, J. Chen, H. Yuan, Kinematics parameter analysis of long jump based on human posture vision, In: *2023 4th International conference on information science, parallel and distributed systems (ISPDS)*, 2023, 349–354. https://doi.org/10.1109/ISPDS58840.2023.10235733
8. S. Huang, M. Gong, D. Tao, A coarse-fine network for keypoint localization, In: *Proceedings of the IEEE international conference on computer vision (ICCV)*, 2017, 3047–3056. https://doi.org/10.1109/ICCV.2017.329
9. Y. Xie, R. Yang, G. Liu, D. Li, W. Wang, Human skeleton action recognition algorithm based on dynamic topological graph, *Comput. Sci.*, **49** (2022), 62–68. https://doi.org/10.11896/jsjkx.210900059
10. M. Wakai, N. P. Linthorne, Optimum take-off angle in the standing long jump, *Hum. Mov. Sci.*, **24** (2005), 81–96. https://doi.org/10.1016/j.humov.2004.12.001
11. A. Shah, D. Prajapati, Recent advances in computer vision and machine learning for athletic performance in jump events, *Augment. Hum. Res.*, **10** (2025), 12. https://doi.org/10.1007/s41133-025-00087-x
12. C. de Boor, *A practical guide to splines*, New York: Springer, 1978. https://doi.org/10.1007/978-1-4612-6333-3
13. R. Cross, The bounce of a ball, *Am. J. Phys.*, **67** (1999), 222–227. https://doi.org/10.1119/1.19229
14. S. Kumar, V. Kanwar, S. K. Tomar, S. Singh, Geometrically constructed families of Newton's method for unconstrained optimization and nonlinear equations, *Int. J. Math. Math. Sci.*, **2011** (2011), 972537. https://doi.org/10.1155/2011/972537
15. L. Breiman, Random forests, *Mach. Learn.*, **45** (2001), 5–32. https://doi.org/10.1023/A:1010933404324
16. B. Ashby, J. Heegaard, Role of arm motion in the standing long jump, *J. Biomech.*, **35** (2002), 1631–1637. https://doi.org/10.1016/S0021-9290(02)00239-7
17. K. Holmquist, B. Wandt, DiffPose: Multi-hypothesis human pose estimation using diffusion models, In: *Proceedings of the IEEE/CVF international conference on computer vision (ICCV)*, 2023, 15931–15941. https://doi.org/10.1109/ICCV51070.2023.01464
18. J. Xu, Y. Guo, Y. Peng, FinePOSE: Fine-grained prompt-driven 3D human pose estimation via diffusion models, In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, 2024, 561–570. https://doi.org/10.1109/CVPR52733.2024.00060
19. S. Mehraban, V. Adeli, B. Taati, MotionAGFormer: Enhancing 3D human pose estimation with a transformer-GCN-former network, In: *Proceedings of the IEEE/CVF winter conference on applications of computer vision (WACV)*, 2024, 6905–6915. https://doi.org/10.1109/WACV57701.2024.00677