scientific reports

Check for updates

OPEN

A deep learning based framework for music-synchronized dance choreography with pose quantization and motion prediction for activity recognition

Weiwei Fan[1,2] & Xuerui An[2]✉

The ability to generate dynamic, expressive dance routines that adapt to various musical compositions has broad applications in activity recognition, performance arts, entertainment, virtual reality, and interactive media, offering new avenues for creative professionals and audiences alike. In this article a deep learning framework is developed for music-synchronized dance choreography through modified vision transformers and graph convolutional networks based on Mexican hat wavelet function for position quantization and motion forecasting. More explicitly high-dimensional pose characteristics are extracted from dance video frames using modified vision transformer to generate a skeletal graph, while modified graph convolutional network captures the spatial and temporal relationships between human joints. The process of discretizing continuous pose data is performed by using K-mean clustering and vector quantized variational autoencoders, respectively. The music synchronization beat-aligned loss was optimized, and the best-tuned weight coefficients were found using two variants of the differential evolution algorithm, based on controlled mutation factors $\mathcal{F}$ =log-sigmoid () and $\mathcal{F}$ =rand(). The proposed architecture with $\mathcal{F}$ =log sigmoid () achieves the lowest Fréchet inception distance (FIDk = 32.451, FIDg = 11.219) and music motion correlation of 0.341 demonstrating enhanced motion synthesis in comparison to existed state of art techniques. The mean fitness value of 6.0294 × 10–10 is obtained with an overall classification accuracy of 97.019% in 0.8431G FLOPs for differential evolution algorithm with $\mathcal{F}$ log-sigmoid (). The framework may be utilized in AI-generated choreography, virtual dance instruction, and interactive entertainment.

Keywords Activity recognition, Human pose estimation, Vision transformers, Graph convolutional networks, Vector quantization, Differential evolution algorithm and monte carlo simulations

Abbreviations

AI	Artificial Intelligence
CNNs	Convolutional Neural Networks
ViT	Vision Transformer-
HPC	High-Performance Computing
GCNs	Graph Convolutional Networks
DTW	Dynamic Time Warping
FFN	Feedforward network
DE	Differential Evolution
ML	Machine Learning
DL	Deep Learning
RNNs	Recurrent Neural Networks
HMMs	Hidden Markov Models
MHA	Multi headed self-attention
τ_0	Output positional encoding

[1]College of Art, Southwest Minzu University, Chengdu 610200, China. [2]Academy of Music and Dance, Aba Teachers University, Wenchuan 623002, China. ✉email: 18633115116@163.com

ξ_f	Predefined frequency
ψ_{MHA}	MHA function
f_1	Fully connected operation
H	Height of the feature map
S	Squeezed value of channel
$\tau_{a,b,p}$	Recalibrated activation
X_{pose}	Pose matrix input
ψ_{SE}	Learned pose embedding vector
STG	Spatio-temporal graph
W_τ	Temporal convolution kernel
$W^{(l)}$	Trainable weight matrix
Z_i	Latent feature
C_k	Pre-defined pose centroid
$f(.)$	Activation function
g_p	Global average pooling
b_0	Bais
τ_{FC}	Fully connected layer
W	Width of the feature map
W_1, W_2	Weight matrices
f_c	Excitation weights of channels
EG	Edges
N	Node of human joint
τ_{CLS}	Outcome of cls-indexing layer
T	Temporal window size
A	Adjacency matrix
Z_q	Quantized pose sate
C_j	Discrete pose clusters

Synchronization of music and choreography is crucial for creating immersive and evocative experiences in film, gaming, virtual reality, and artificial intelligence (AI) generated dance that is an industry of around $31.2 billion[1]. It harmonizes motion dynamics with rhythmic patterns, enhancing audience perception and retention by 70%. AI-powered choreography tools utilizing beat tracking algorithms and motion prediction models automate dance creation, enhancing music video production efficiency by 50%[2]. Notably, platforms such as TikTok and YouTube rely heavily on coordinated music-video trends, with short-form video consumption increasing by 55% per year that show the importance of the subject. Moreover, with the rise of virtual concerts, metaverse performances, and AI-driven content creation, music synchronization enhances engagement, profitability, and innovation. Although in the past many traditional techniques were used for this purpose, they were time-consuming and exhibited low efficiency in terms of musical smoothness and pitch quality[3,4]. In the last half decade lots of advance form of deep learning has been introduced by the researchers involving convolutional neural networks (CNNs), explainable AI and vision transformer (ViT) etc. Quantifying music-dance[5–7].

ViT is a deep learning architecture that use the self-attention mechanism of transformers to analyze images, in contrast to CNNs. CNNs utilize local receptive fields, whereas ViT segment an image into fixed-size patches, flatten them, and treat them as tokens[8]. It encapsulates global interdependencies among image regions by embedding each patch into a high-dimensional vector and transmitting it through multi-head self-attention layers. ViT excels in contextual understanding tasks such as image categorization, object detection, and pose estimation due to its preservation of spatial relationships among patches through positional encodings. ViT surpasses CNNs on extensive datasets by identifying intricate patterns with reduced inductive biases. It requires substantial processing power and an extensive training dataset to generalize, making it most suitable for high-performance computing (HPC)[5]. Keeping in view synchronized dance choreography with pose quantization and motion prediction, the idea of a special CNN based on the graph theory can be one of choice. Thes graph convolutional networks (GCNs) are deep learning architectures capable of extracting spatial information from structured datasets such as human skeletons as shown in Fig. 1, social networks, and three-dimensional meshes[9]. It implements convolutions on graph topologies, with nodes symbolizing items like human joints and edges denoting their interactions. It effectively captures spatial dependencies and hierarchical structures through the use of graph convolution and the adjacency matrix. GCNs are proficient at deriving spatial information from skeletal graphs for human pose estimation, motion forecasting, and action identification. AI-driven choreography, robotics, and intelligent motion analysis leverage their ability to depict intricate spatial interactions[10].

While ViTs and GCNs provide the theoretical backbone for visual and structural data processing, recent advancements have further diversified these architectures into specialized models that tackle complex tasks across vision, robotics, and multimodal learning domains[11–13].Progress in computer vision has been evident through deformable 3D convolutional super-resolution techniques and robust interaction recognition using filtering methods[14,15], complemented by multisensory locomotion and skeleton-based spatio-temporal GCN-transformers for enhanced human activity analysis[16,17]. Robotics has similarly benefited, where stiffness identification through optimal pose selection has strengthened mechanical precision[18], and biomedical applications now leverage graph neural networks for drug repositioning alongside robust neural solvers for complex-valued optimization[19,20]. Security and adversarial robustness have been advanced via transferable attack strategies[21], while rehabilitation technologies evolved through adaptive exoskeleton control strategies for flexible assistance and multimodal gait planning[22,23]. Generative approaches have expanded through fine-grained motion generation supported by large language models[24], with probabilistic modeling enhanced by

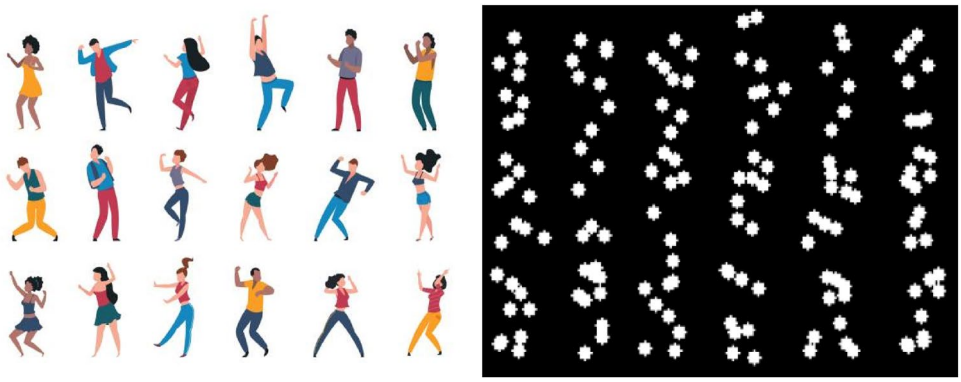

Fig. 1. Extraction of structured dataset for GCNs.

Syms	Explanations	Syms	Explanations
γ_1, γ_2 and γ_3	Best tuned weight coefficients	τ_0	Outcome of the positional encoding
ξ_f	Predefined frequency	d_H	Head dimensionality
τ	Residual connection	H	Height of the feature map
L	Beat aligned loss	W	Width of the feature map
f_1	Fully connected operation	S	Squeezed value of channel
$f(.)$	GeLU activation function	w_1, w_2	Reall valued network weights
b_0, b_1	Biases	$\tau_{a,b,p}$	Recalibrated activation
f_c	Excitation weights of channels.	τ_{CLS}	Outcome of indexing layer,
W_τ	Temporal convolution kernel	τ_{FC}	Output of fully connected layer
T	Temporal window size	$\bar{\bar{A}}$	Adjacency matrix
$H^{(l)}$	Feature matrix at layer l	$\bar{\bar{D}}$	Diagonal degree matrix
$W^{(l)}$	Trainable weight matrix of the layer l	g_p	global average pooling operation

Table 1. List of mathematical symbols (Syms) with their corresponding definitions.

adaptive Bayesian curve fitting[25]. In parallel, visual odometry has been optimized through adaptive keypoint extraction[26], while transformer-based IoT sequence models addressed temporal bias and non-stationarity[27]. Creative applications emerged with adversarially enhanced diffusion models for facial attribute synthesis[28], and surveys on hallucination in large language models revealed critical challenges for trustworthy AI deployment[29].

For effective dance choreography, motion prediction, and action identification, K-Means clustering discretizes continuous human motion into a limited number of representative poses[30]. Pose feature vectors, such as joint positions, angles, and velocities, are clustered into k groups using this method; each center represents a quantized pose state. K-Means assigns poses to the closest centroid by updating cluster centers until convergence using Euclidean distance minimization[31]. It clusters similar poses to create structured motion sequences for AI-driven choreography and gesture synthesis, increasing computational efficiency and real-time applicability in motion analysis systems[32,33]. The proposed architecture offers numerous major advancements. Our method embeds SE modules in Vision Transformer blocks to improve channel-wise attention and suppress unnecessary features in early layers, unlike CNN or transformer-based methods. Graph Convolutional Networks modified with the Mexican Hat Wavelet function (GCN-MHW) capture spatio-temporal interactions in skeletal graphs more expressively and outperform ordinary GCNs in dynamic pose simulation. A new differential evolution variation with a log-sigmoid mutation factor increases synchronization precision and optimization stability, especially under the beat-aligned loss function. Three components comprise a tightly connected pipeline for high-fidelity, music-synchronized choreography. ViT offers a baseline for learning spatial properties from pose sequences. By adding temporal attention, TimeSformer expands on ViT and is especially useful for simulating dancing moves over long periods of time[34]. In order to enable scalability to larger motion sequences, Performer and Linformer have also been used to lower the quadratic complexity of attention. In order to jointly simulate temporal dependencies and skeleton graph structures, hybrid techniques that include transformers and GCNs have been studied more recently[35,36].

The symbols used in the mathematical relations of the proposed architecture is provided below in Table 1 for the study carried out.

The primary contributions of the proposed study are outlined as follows:

- The development of a ViT based on squeeze and excitation (SE) properties which embed the attributes of SE module in each transformer block that dynamically recalibrate channel-based feature maps.
- The formulation of a modified GCN deep learning (DL) model based on Mexican hat wavelet (MHW) function named as GCN-MHW model with temporal convolution window and kernel.
- The music synchronization beat aligned loss was optimized and found best tuned weight coefficients γ_1, γ_2 and γ_3 using two variants of differential evolution algorithm based on controlled mutation factor $\mathcal{F} = logsigmiod()$ and $\mathcal{F} = rand()$.
- The outcomes of the proposed framework are evaluated for the AIST++ dataset with a motion quality factor of 35.451 and 11.219 for kinetic and geometric features, respectively as referenced in[37] with a beat alignment score of 0.341 and other sate of art method[37,38].
- The reliability, stability, and computational complexity of the proposed DL architecture is examined based on the randomized numerical simulation based on sufficient independent runs, risk analysis and drawing the surface of computational budget.

The subsequent sections of the article delineate the pertinent state of the art and documented findings in Sect. "Related work and state of art", along by a description of the dataset and its characteristics. Section "Material and methods" describe the proposed deep learning framework as well as the variants of differential evolution algorithm. Section "Results and discussion" presents the parameter values & setting exploited in the simulation of deep learning architecture for synchronization of music with the pose and motion prediction, software hardware specifications, results, their analysis and a comparison with current state-of-the-art methodologies. The concluding section outlines the conclusions and proposes avenues for future investigation.

Related work and state of art

Traditional music synchronization employed manual feature extraction and rule-based alignment. It utilizes chroma-based representations to align various renditions of the same composition by recording pitch class distributions across time[39]. Similarly, dynamic time warping (DTW) was frequently employed by Schramm et al.[40], to align chroma characteristics in order to address variations in musical speed and timing while the hidden Markov models (HMMs) represented the temporal dependencies of musical sequences and facilitated synchronization. These methods proved effective in structured music but were less resilient in complex, polyphonic, or loud compositions. In 1994, Goto et al., employed conventional technique based on onset detection and beat tracking to synchronize tempo by detecting percussive or harmonic alterations in the musical input[41], the proposed system correctly tracked beats with an accuracy of 90% (27 out of 30) commercially distributed popular songs. Moreover, the tempo estimation and musical sequence alignment frequently employ Fourier-based techniques[42] and autocorrelation functions. Another approach is Score-based synchronization that employed artisanal matching techniques to accurately map note events. Nonetheless, these methodologies necessitated domain expertise and were susceptible to performance style, instrument timbre, and background noise, hence constraining their applicability across musical genres.

Another important aspect of the music domain is dance choreography, which traditionally relied on manual notation and expert design. In the age of 1970's choreographers utilized Labanotation and Benesh Movement Notation to document and standardize dance movements for accurate invention, analysis, and reproduction[43] that is quite evident from the work of Watts.et.al, These systems depicted bodily locations, movement paths, and temporal aspects using symbols and structures. Choreographers employed improvisation and repeated refinement to modify sequences in accordance with creative intent and musical beat alongside dancers[44]. Classical ballet and folk dances employed uniform movement vocabularies transmitted through direct instruction and oral tradition. Symmetrical and aesthetically pleasing formations were occasionally created utilizing mathematical models such as Lissajous curves and geometric patterns. These tactics were effective in structured dance genres but lacked adaptability in modern and experimental choreography[35,45]. Manual choreography was labor-intensive and reliant on the skill levels of dancers and instructors, necessitating extensive rehearsal and interpersonal engagement[46]. Keeping in view the difficulties, applicability and weaknesses of the tradition music and choreography synchronization the need of the machine learning techniques arises that can reduce the human error and can perform the task in very less time.

In this regard, simulated annealing (SA), and genetic algorithms (GA) are expolited by many researchers[47,48] to optimize movement alignment and sequencing music synchronization, dance choreography, posture quantization, and motion prediction by employing similarity metrics. These algorithms systematically modify posture sequences to minimize synchronization errors, hence enhancing dance motions as it has inherent capacity for natural selection to generate optimal movement patterns. They also enhance dance choreography by systematically reducing randomness in the exploration of the search space, thereby converging on an optimal movement sequence that aligns most effectively with the music's tempo, beat, and style with an accuracy of (83 – 35) % while the beat hit rate is found to be (85–90) %. DL has revolutionized music synchronization by aligning rhythms and motions that comprehend temporal relationships between audio features and human actions by utilizing extensive datasets, in contrast to rule-based methods. Recurrent Neural Networks (RNNs)[49,50], Long Short-Term Memory networks (LSTMs)[5], and transformers have been extensively employed to analyze time-series data from music and dance motion sequences to guarantee seamless synchronization while the classes of dances are limited like four only. Spectrogram-based CNNs[7] extract harmonic and percussive components to enhance beat tracking and align dancing movements with musical patterns with an accuracy of 89.6% for the beat hit rate. Real-time reinforcement learning (RL)[51,52] models enhance choreography synchronization and adjust to variations in speed and style. These models can create highly coordinated dance sequences that match complex musical compositions by merging deep learning with audio, video, and motion capture data. Moreover, the methods like spatio-temporal graph convolutional networks (ST-GCNs)[34,53], variational autoencoders (VAEs)[54]

and generative adversarial networks (GANs)[8,55] have been employed to synthesize diverse and fluid dance movements from extensive human motion datasets to enhance pose estimation and motion prediction. These DL algorithms enable AI-driven choreographic systems to produce dynamic, emotive, and highly synchronized dance performances, exceeding rule-based approaches.

Li. Ruilong et al.[37] presents the AIST + + dataset with 3D dance motion covering 10 genres with a sufficient large number of video clips, it applies transformer based deep learning followed by feedforward layers to get a motion quality factor of 35.35 and 12.40 for kinetic and geometric features, respectively. The motion music correlation was performed automatically with a beat alignment score of 0.241. Another leading contribution observed in the literature survey is performed by Li. et., al in 2020[38] that exploits DL to generate diverse dance motions with transformer architecture for synthesis. It exploits two steam motion transformers capable enough for capturing long term dependencies and conditions to get motion quality factor of 86.43 and 20.58. for kinetic and geometric features, respectively. Keeping in view the limitations, strengths and weakness of various traditional as well as modern ML techniques from the literature, it is clear that the need of robust, reliable and comprehensive architecture for music-synchronized dance choreography with pose quantization and motion prediction is always required for a highly dynamical and profitable industry. Moreover, in built dance classification with a reasonably high accuracy is also required.

In this regard, a deep learning architecture is formulated that is based on the blend of squeeze and excitation ViT, GCN activated through Mexican hat wavelet function and pose quantization by k-mean algorithm as well as vector quantized variational autoencoders. The music synchronization beat aligned loss is optimized and find best tuned weight coefficients γ_1, γ_2 and γ_3 using two variants of differential evolution algorithm based on controlled mutation factor $\mathcal{F} = logsigmiod()$ and $\mathcal{F} = rand()$. The outcomes of the proposed framework are evaluated for the AIST + + dataset keeping in view the performance measures like motion quality, motion diversity, beat alignment score and overall classification accuracy. The reliability, stability, and computational complexity of the proposed architecture is examined based on the randomized numerical simulation based on sufficient independent runs, risk analysis and drawing the surface of computational budget.

Materials and methods

This section is divided into four main parts in the first part the detail about the AIST + + dataset is presented along with its associated difficulties. In the second part modified ViT is presented along with its mathematical details. This part also explains the mathematical aspects of proposed GCN-MHW model. The third part depicts the details regarding two proposed variants of the differential evolution algorithm used for the training of the proposed architecture. The performance parameters are presented in the fourth part of this section. The overall workflow of the proposed architecture is presented in Fig. 2.

Dataset description

AIST + + dance video dataset is considered to be the one of the tedious and large-scale 3D human dance pose synchronization and motion estimation dataset that has 9 views of camera intrinsic and extrinsic parameters with 10 different dance styles associated with the movement style and music beat. These genres encompass break (BK), pop (PP), lock (LK), hip-hop (HH), house (HO), waack (WK), krump (KP), ballet jazz (BAZ), contemporary (CT), and freestyle (FES). 1408 clips totaling 10.1 h of dancing motion, featuring diverse moves, were recorded at 60 frames per second for motion data and 30 frames per second for video data. The motivation behind the use deep learning architecture come due to the unique movement of pattern in the genres, temp and beat synchronization in each genres make a complex problem. The training is applied on the 80% of the random data selected while rest of the 20% dataset is used for testing, moreover it is worth mentioning that each dance genre comprises 85% basic choreographies and 15% advanced choreographies. Along with the traits of the view, images, genres, subject and sequence, the dataset al.so attributes like music, 3D joint pose, 3D joint rotation and 2D key points unlike the other existed datasets in the literature like AMASS[56], GrooveNet[57,58], EA-MUD[58] etc.

Enhancing the generalizability of the suggested architecture is significantly aided by the diversity of the AIST + + dataset. Each of the ten dance styles included in the dataset—including hip-hop, house, ballet jazz, modern, and freestyle introduces different body dynamics, rhythmic patterns, and degrees of motion complexity. The model may learn heterogeneous motion distributions through exposure to this stylistic variability, which enables it to adjust to dancing styles that are not visible during inference. In addition, the collection offers recordings from nine distinct camera perspectives with a range of intrinsic and extrinsic factors. This multi-view configuration ensures consistent performance under various visual situations by strengthening the model's resistance to viewpoint shifts. Together, the genre diversity and viewpoint variation enhance the capacity of the proposed framework to generalize beyond training data and generate realistic, beat-synchronized choreography under diverse scenarios.

The proposed architecture

The proposed framework integrates SE-ViT, GCN-MHW, and two variants of pose quantization. The mathematical formulation of each key component is detailed to ensure reproducibility. Vision Transformer (ViT)[59] has demonstrated strong adaptability in medical imaging, with numerous variants tailored for diverse scientific and technological domains. Our architecture leverages a squeeze-and-excitation enhanced ViT (SE-ViT), where SE modules are embedded within each transformer block to adaptively recalibrate channel-wise feature responses. Specifically, after each transformer block, the SE module performs global average pooling to aggregate spatial information, followed by channel-wise weighting to emphasize informative features while suppressing redundant ones. Then, using attention weights, scale the features maps, effectively expanding the most relevant channels and suppressing irrelevant ones. The proposed SE-ViT transformer accepts the input of $384 \times 384 \times 3$. The initial layer is patch embedding is employed with 16×16 patch sizes and apply position

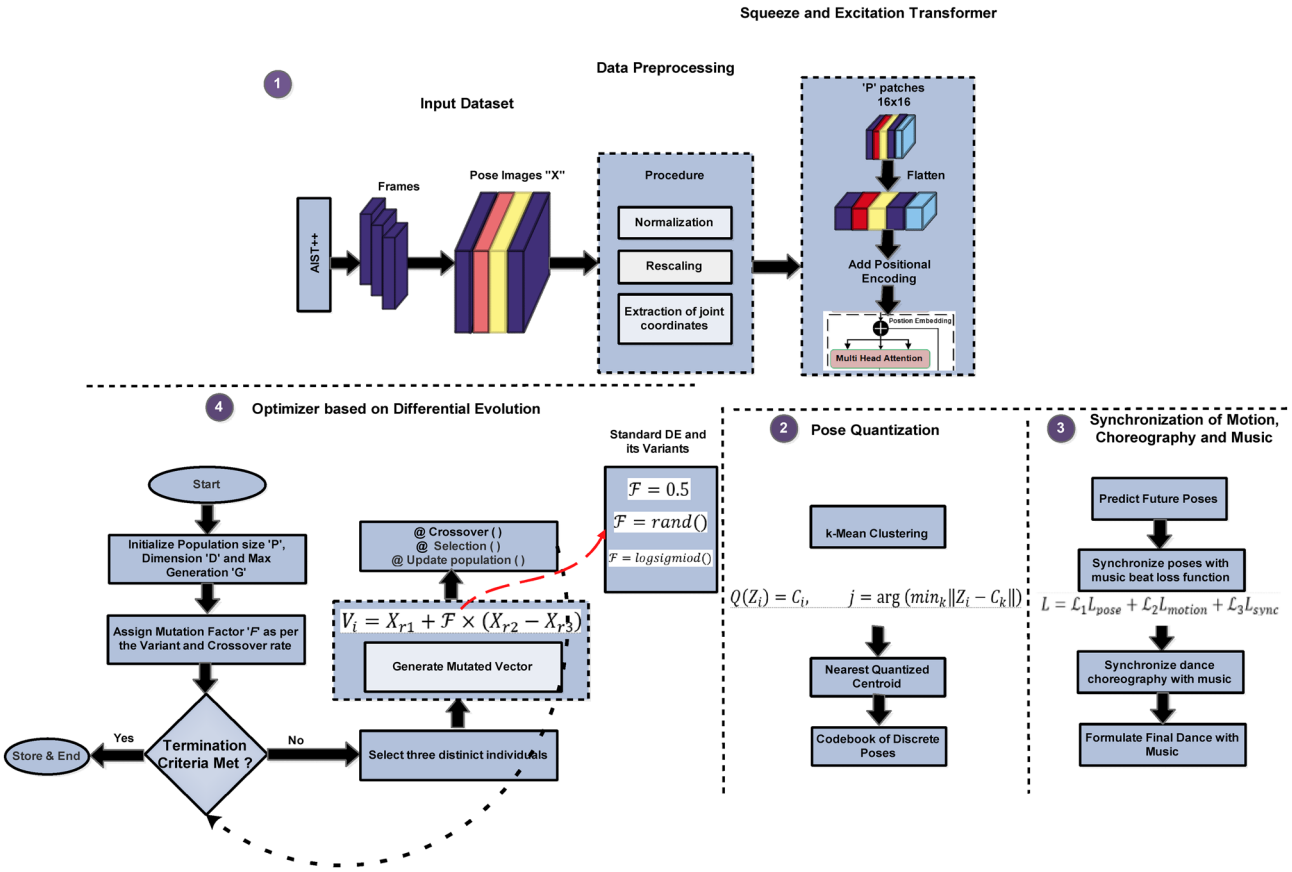

Fig. 2. Graphical workflow of the proposed architecture, illustrating the sequential data flow, major processing modules, and their interconnections.

encoding to retain the spatial information. This positional encoding is added element wise to the patches and is represented as given in Eq. (1). The equation is introduced in the SE-ViT component. Vision Transformers treat an image as a sequence of patches. Since transformers were originally designed for sequential data (e.g., language), they lack inherent spatial awareness. Therefore, positional encoding is crucial to embed spatial structure.

$$\left.\begin{array}{c} \tau_0\,(new) = \tau_0\,(old) + \xi_n \\ \xi_n\,[i] = sin\,(2\pi\,f, i)\,, cos(2\pi\,f, i)(\xi_f,\ 2\pi\,f) \end{array}\right\} \tag{1}$$

where τ_0 is the outcome of the positional encoding and ξ_f is the predefined frequency. The spatial links between patches are lost during flattening, hence positional encoding is necessary. A fixed, non-trainable spatial reference in the sinusoidal formulation lets the model identify patch positions and preserve global context during attention calculation.

The first transformer is started by employing multi headed self-attention (MHA) layer with 3 heads to perform self-attention across each patch in order to learn the long-range relationship who mathematical relationship is given in Eq. (2) that governs the multi-head attention used in the transformer layers. It's applied to allow the model to attend to different parts of the input image simultaneously, capturing long-range dependencies.

$$\psi_\delta\ (A,\ B,\ G) = Softmax\left(\frac{AB^\tau}{\sqrt{d_h}}\right) G, \psi_{MHA}\,(\tau\,) = \oplus\ (H_1, H_2, H_3)\, w_0, \quad w_0 \in \mathbb{R}^{(H.d_H)\times d} \tag{2}$$

where ψ_{MHA} is the outcome of MHA. The scaled dot-product attention technique calculates attention weights from queries and keys, scaling by head dimensionality d_H. Concatenating the outputs from each attention head preserves multi-scale relationships among patches. After that, a residual connection is added, followed by the normalization layer that is defined as:

$$\tau' = N(\psi_{MHA}\,(\tau\,) + \tau\,) \tag{3}$$

This step applies a residual connection followed by layer normalization standard in transformer architectures to stabilize training. Stabilizing the learning process via a residual link and normalization helps gradient flow and

prevents deep network gradient vanishing. In addition, feed forward network (FFN) procedure is performed on the outcome of MHA layer. This process is consisting of two linear layers with nonlinear activation used as GeLU function that is formulated as an additional layer and is defined as:

$$\psi_{FFN} = f(\tau w_1 + b_0) w_2 + b_1, \quad w_1 \varepsilon \mathbb{R}^{d \times d_f} \text{ and } w_2 \varepsilon \mathbb{R}^{d_f \times d} \tag{4}$$

where $f(.)$ is the GeLU activation. The feed-forward network (FFN) processes attended features using two linear transformations and a non-linear GeLU activation. Deepening the network helps it capture complex channel transitions. After first transformer, SE block is performed to apply channel wise attention to the feature maps for enhancing the most relevant features. The SE module contains the global average pooling, fully connected and sigmoid activation. The mathematical modeling of SE block is defined in Eq. (5) as follows:

$$\psi_{SE} = \begin{cases} g_p = \frac{1}{H,W} \sum_{a=1}^{H} \sum_{b=1}^{W} \tau_{a,b,S} \\ f_1 = \phi\ (W_2.f\,(W_1.S)) \\ \tau_{a,b,p}.f_c \end{cases} \tag{5}$$

where g_p is the global average pooling operation and H, W is the height and width of the feature map, f_1 is the fully connected operation, S is the squeezed value of channel. The SE block rebalances feature channels by stressing valuable features and suppressing unhelpful ones. Global average pooling aggregates spatial information, followed by channel-wise weighting through fully linked layers and non-linear activations (ReLU and Sigmoid). W_1, W_2 are the weight matrices of the fully connected layer, ϕ , f is the sigmoid and ReLU activation, respectively. $\tau_{a,b,p}$ is the recalibrated activation and f_c is the excitation weights of channels. After this, posture embedding is added to capture the global representation and then employed classification head by using Eq. (6).

$$\psi_{SE} = \begin{cases} \tau_{CLS} = \tau\ [0] \\ \tau_{FC} = W.\tau_{CLS} + b \\ \mathbb{C}_k = \frac{e^{\tau_i}}{\sum_{i=1}^{N} e^{\tau_i}},\ i \in [1, N] \end{cases} \tag{6}$$

where τ_{CLS} is the outcome of cls-indexing layer, τ_{FC} is the fully connected layer that passed to the softmax activation for final output $\mathbb{C}_k$. Finally, a classification layer receives SE-transformed feature output. To predict class distribution, the token is extracted, linearly projected, and softmaxed. For graph-based processing, the GCN-MHW module receives this posture embedding. The architecture of SE based VitT is presented in Fig. 3.

The SE-ViT proposed architecture in Fig. 3 takes the pose video frames as input (X_{pose}) and extract joint embeddings in the form of deep pose features. This input image is divided into patches and processed through self-attention layers as provided in SE-ViT.

$$\psi_{SE} = SE - ViT\,(X_{pose}) \tag{7}$$

where ψ_{SE} is the learned pose embedding vector.

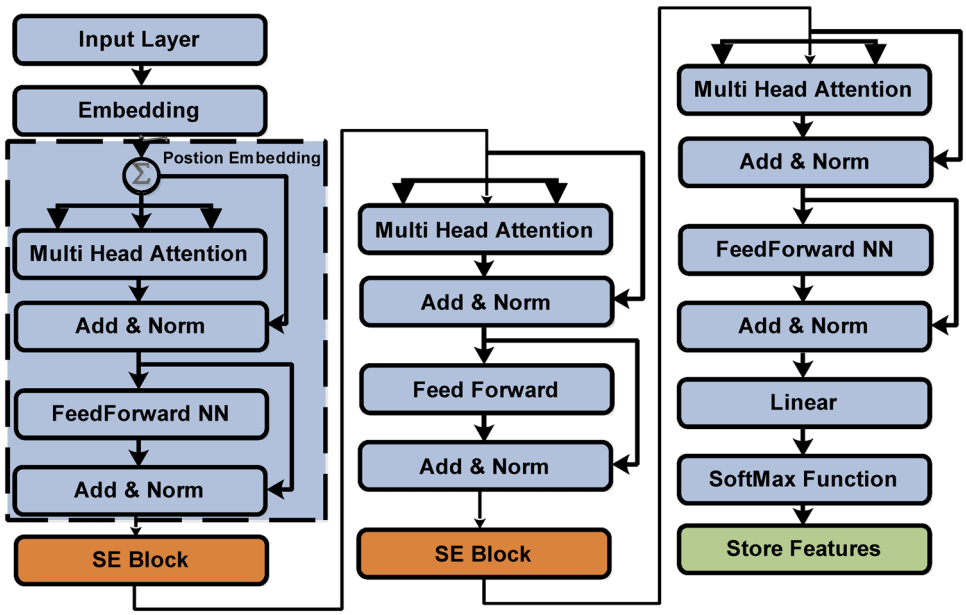

Fig. 3. Proposed SE-ViT framework, showing integration of SE modules within transformer layers.

Now considering the human skeleton that can be represented by the spatio-temporal graph[60] as $STG = (N,\ EG)$ where N in graph is nodes that is represented by human joints and EG are the edges represented as connection between the joints. The connectivity among the joints is represented by a mesh of interconnections take as adjacency matrix A that models the spatial connectivity of joints is given below:

$$A_{i,j} = \begin{cases} 1, & if\ joints\ i\ and\ j\ are\ connected \\ 0, & otherwise \end{cases} \tag{8}$$

The connection between the signal and graph is established through a feature vector in which each joint i at frame t is represented as:

$$X_i^t = \left[x_i^t,\ y_i^t,\ z_i^t, v_i^t\right] \tag{9}$$

where the parameters $x_i^t,\ y_i^t,\ z_i^t$ and v_i^t explain 3D coordinates of the joint and motion related to the features. Therefore, the full feature matrix with N number of joints in a frame having d feature dimension is give as $X^t \in \mathbb{R}^{N\times d}$. The graph convolution operation updates node features based on neighbors.

$$H^{l+1} = \psi\left(\bar{\bar{D}}^{-\frac{1}{2}}\bar{\bar{A}}\bar{\bar{D}}^{-\frac{1}{2}}H^{(l)}W^{(l)}\right) \tag{10}$$

where $H^{(l)}$ is the feature matrix at layer l, $\bar{\bar{A}} = A + I$ is the adjacency matrix with self-loops with $\bar{\bar{D}}$ is the diagonal degree matrix such that $D_{ii} = \sum_j A_{ij},\ W^{(l)}$ is the trainable weight matrix of the layer l having σ an activation function take as Mexican hat wavelet function. Because of its band-pass properties, which highlight mid-frequency information while attenuating both low- and high-frequency noise, the Mexican Hat wavelet function is used as the activation function in the suggested GCN-MHW. This characteristic keeps the global structural interdependence of the human skeleton intact while enabling the network to capture dynamic local variations in joint movements. The model can better describe rhythmic and oscillatory motion patterns that naturally correspond with musical beats because to the Mexican Hat wavelet's capability for multi-scale feature representation, in contrast to traditional activations like ReLU or sigmoid that only provide pointwise nonlinearities. Additionally, its filtering effect makes motion predictions smoother and more reliable while improving robustness against noisy pose data.To provide a clearer understanding of the graph convolution process, we elaborate on the structure and dynamics of the skeleton graph. Each input frame is represented as a set of joints, where each joint *i* at time *t* is encoded by a feature vector $X_i^t = [x_i^t, y_i^t, z_i^t, v_i^t]$ as shown in Eq. (9), comprising its 3D spatial coordinates and velocity. These joint-level features are organized into a matrix $X_t \varepsilon \mathbb{R}^{N\times d}$, where *N* is the number of joints and *d* is the dimensionality of the features.

The spatial relationships between joints are captured using an adjacency matrix *A*, where A_{ij}=1 if joint *i* is directly connected to joint *j* in the skeletal structure, and 0 otherwise. The graph convolution operation updates the representation of each joint by aggregating feature information from its neighbors, normalized by the degree matrix *D*, as shown in Eq. (10). This process allows the model to learn spatial dependencies and local structural patterns in the human body. Considering the definition of $\psi\ (t)$ as $\psi\ (t) = \frac{1}{sqrt(2\pi\ )\ \sigma^3}\left(1 - \frac{t^2}{\sigma^2}\right)\exp\left(-\frac{t^2}{2\sigma^2}\right)$, therefore, the relation of the Eq. (10) will transform as given in Eq. (11).

$$H^{l+1} = \frac{1}{sqrt\ (2\pi\ )\ \sigma^3}\left(1 - \frac{\left(\bar{\bar{D}}^{-\frac{1}{2}}\bar{\bar{A}}\bar{\bar{D}}^{-\frac{1}{2}}H^{(l)}W^{(l)}\right)^2}{\sigma^2}\right) \times \exp\left(-\frac{\left(\bar{\bar{D}}^{-\frac{1}{2}}\bar{\bar{A}}\bar{\bar{D}}^{-\frac{1}{2}}H^{(l)}W^{(l)}\right)^2}{2\sigma^2}\right) \tag{11}$$

The time evolution of poses is modeled by temporal convolution $H_t^{l+1} = \sum_{\tau=-T}^{T} W_\tau\ H_{t+\tau}^{(l)}$ where W_τ is the temporal convolution kernel, T is temporal window size. This operation captures motion transitions over time making it crucial for dance choreography. The final pose sequence $\widehat{X}^{t+1}$ is predicted using

$$\widehat{X}^{t+1} = f_{GCN-MHW}(X^t, A) \tag{12}$$

where $f_{GCN-MHW}$ is the trained GCN-MHW model. For music synchronization beat aligned loss is introduced as:

$$L = \gamma_1 L_{pose} + \gamma_2 L_{motion} + \gamma_3 L_{sync} \tag{13}$$

where L_{pose} is the mean square error between the predicted and ground-truth poses, L_{motion} motion continuity loss that is difference between two frames, L_{sync} best synchronization loss aligning poses with music tempo. γ_1, γ_2 and γ_3 are weight coefficients.

Once GCN-MHW learn spatial and temporal relationships in skeletal graphs, the process of discretizing continuous pose data is performed by using K-mean clustering[61,62] and vector quantized variational autoencoders (VQ-VAE)[62], respectively. Therefore, the latent embedding representation of the pose is given below:

$$Z = GCN - MHW(X, A) \tag{14}$$

If Z_i is the latent feature of a pose, then applying the K-mean clustering the quantized data will be calculated as:

$$Q(Z_i) = C_i, \qquad j = \arg(min_k \parallel Z_i - C_k \parallel) \tag{15}$$

where C_k are pre-defined pose centroid and the function assigns each pose Z_i to the closest centroid.

The VQ-VAE is formulated as below:

$$Z_q = \arg(min_j \parallel Z - C_j \parallel) \tag{16}$$

where Z_q is the quantized pose sate and C_j are the discrete pose clusters. This transforms continuous nose embedding into a discreate representation.

The training, validation and inference phases are emphasized in Fig. 4, where dance video frames are translated into skeletal position sequences during training, and music audio is converted into Mel-spectrogram characteristics. The GCN-MHW module processes these inputs to capture spatiotemporal dependencies in the skeletal graph, while the SE-ViT module extracts pose embeddings. K-means clustering and vector-quantized variational autoencoders are used to quantize the final features, which are subsequently optimized via differential evolution algorithm variations under a beat-aligned loss function. To direct hyperparameter adjustment, the validation step assesses beat alignment, motion diversity (Dist), and motion quality (FID). Lastly, during the inference phase, the trained model creates realistic skeletal motion sequences and categorizes dance styles by creating beat-synchronized dance choreography from unseen audio inputs.

The proposed architecture based on SE-ViT, GCN-MHW and two variation of pose quantization is given below in the form of psedocode:

Input: Music audio clip A, Pose sequence $P = \{p_1, p_2, ..., p_T\}$

Output: Beat-aligned predicted pose sequence $\hat{P} = \{\hat{p}_1, \hat{p}_2, ..., \hat{p}_T\}$

Extract Mel-spectrogram features from audio clip A

Divide audio spectrogram into T audio tokens $\{a_1, a_2, ..., a_T\}$

for t = 1 to T do

    Extract keypoints from pose frame p_t using OpenPose

    Form token $\tau_t = [a_t \oplus p_t]$ via early fusion

end for

Pass tokens $\tau = \{\tau_1, ..., \tau_T\}$ through SE-ViT encoder

    Compute positional encoding and apply multi-head attention

    Apply residual connections, FFN, and SE recalibration

Obtain encoded embeddings $Z = \{z_1, ..., z_T\}$

Construct skeleton graph G = (V, E) from pose sequence

for each node $v_i \in V$ do

    Update node features using GCN-MHW:

    $h_i \leftarrow \sigma(\sum\{j \in N(i)\} w_{ij} \cdot MHW(z_j))$

end for

Apply DE-based optimization on decoded motion embeddings

Minimize synchronization loss L_{sync} between audio beat and motion rhythm

Output optimized predicted pose sequence $\hat{P} = \{\hat{p}_1, ..., \hat{p}_T\}$

Proposed varients of differential evolution

Differential Evolution (DE) is a population-based stochastic optimization algorithm designed for continuous, nonlinear, and high-dimensional optimization problems[63]. The method operates through iterative processes of mutation, crossover, and selection, progressively refining a candidate population toward the global optimum. Unlike gradient-dependent techniques, DE is particularly effective for complex or noisy black-box optimization tasks since it does not require derivative information[64]. The procedure begins with the random initialization of a solution population, after which the mutation operator generates trial vectors by adding the scaled difference of two population members to a third individual[65]. In DE, the crossover operation fuses mutant vectors with target solutions, thereby maintaining population diversity and facilitating exploration of the search space. This phase is followed by the selection mechanism, which ensures that only the fittest individuals, i.e., those yielding superior objective function values, are retained for the subsequent generation[66]. Such a survival-of-the-fittest strategy drives the evolutionary process toward optimality. The inherent strengths of DE

namely, robust performance across diverse landscapes, fast convergence rates, and resilience against premature entrapment in local optima have established its prominence in domains such as machine learning hyperparameter tuning, engineering optimization, and control system design. Moreover, advanced variants like Adaptive Differential Evolution and Hybrid Differential Evolution (HDE) introduce dynamic adaptation of key control parameters (mutation factor *F* and crossover rate *CR*), thereby improving convergence stability and computational efficiency in highly complex optimization scenarios[67]. The pseudocode of exploited for the variants of DE is as follows:

Input: Initial population $X = \{x_1, x_2, ..., x_N\}$, mutation factor $\mathcal{F}$
Output: Optimized pose embeddings minimizing beat-aligned loss
Initialize population X of candidate pose sequences
Set mutation strategy $\mathcal{F}$ = log-sigmoid ($\alpha \cdot$ rand ())
for generation g = 1 to G-max do
  for each target vector x_i in X do
    Randomly select three distinct vectors $x_a, x_b, x_c \in X$
    Generate mutant vector:
    $v_i = x_a + \mathcal{F} \cdot (x_b - x_c)$
    Perform crossover to create trial vector u_i:
    $u_{ij} = v_{ij}$ if rand() < CR else x_{ij}
    Evaluate fitness $f(u_i)$ and $f(x_i)$
    if $f(u_i) < f(x_i)$ then
    $x_i \leftarrow u_i$ // Replace with better candidate
    end if
  end for
end for
Return best candidate x_{best} minimizing beat-aligned loss L_{sync}

$$V_i = X_{r1} + F \times (X_{r2} - X_{r3}) \tag{17}$$

where $\mathcal{F}$is the mutation factor that controls the differential weights and create diversity. The usually value of $\mathcal{F}$ is taken as 0.5. However, two variants are purposed based on the different value of $\mathcal{F}$. .

Variant-I The $\mathcal{F} = rand()$ function introduces stochasticity to the search space, hence aiding optimization algorithms in diversifying, circumventing local optima, and investigating a broader array of possible solutions therefore, it produces random perturbations to mitigate early bias in search outcomes.

This randomization enables algorithms to investigate many sections of the search space, thereby reducing premature convergence to local optima and enhancing the probability of attaining the global optimum. Therefore, the relation of Eq. (17) can be written as:

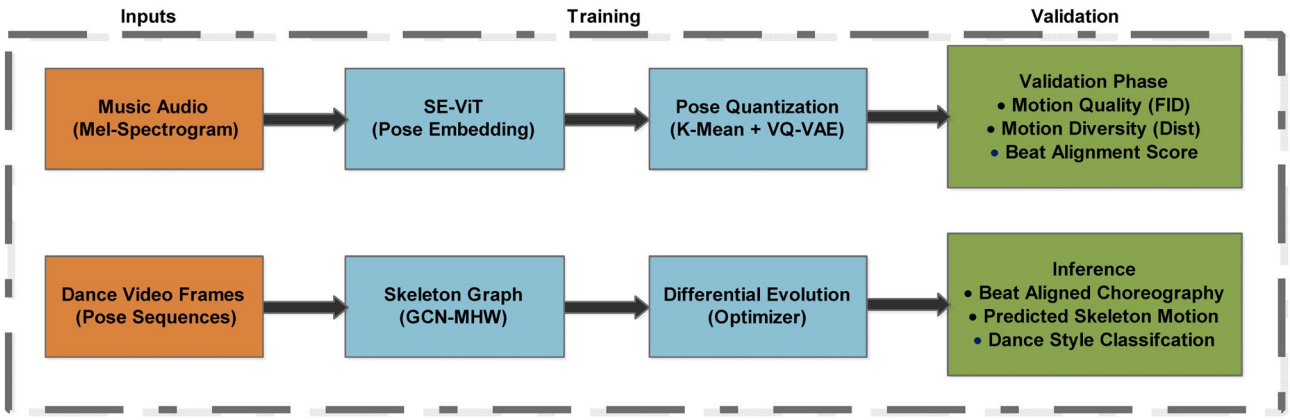

Fig. 4. Proposed training, validation, and inference phases.

$$V_i = X_{r1} + rand() \times (X_{r2} - X_{r3}) \tag{18}$$

Variant-II The second modification is based on introducing log-sigmoid function $\mathcal{F} = logsigmiod()$ in Eq. (17) that would influences search space and prevent numerical instability and inefficient optimization. It mitigates outliers by compressing large input values towards the upper limit and small values towards the lower limit, facilitating smoother convergence in evolutionary algorithms. Therefore, the relation in Eq. (17) will be formulated as.

$$V_i = X_{r1} + \frac{1}{1 + \exp(X_{r2} - X_{r3})} \tag{19}$$

Due to its non-linear and continuous nature, output variations occur gradually, thereby averting sudden shifts that could hinder optimization, moreover, normalization and limiting outputs will produce robustness and efficiency of DE algorithm in addressing complicated, non-linear, and high-dimensional problems by regulating the search space. The mutation factor $\mathcal{F}$ significantly impacts the convergence and stability of the DE algorithm. The variation with $\mathcal{F}$ =rand() increases stochasticity, allowing the optimizer to explore a wider search space and avoid local minima.

Results and discussion

The proposed architecture has been rigorously evaluated on a high-performance computational setup specifically configured to support deep learning experiments. The hardware environment comprises an Intel® Core™ series processor coupled with 128 GB of DDR4 RAM, which ensures high-capacity parallel data processing and memory-intensive computation. For storage, a 256 GB solid-state drive (SSD) is integrated for rapid read–write operations, complemented by a 2 TB hard disk drive (HDD) to accommodate large-scale datasets and experimental logs. Model training and inference are accelerated by an NVIDIA RTX GPU with 8 GB of dedicated VRAM, enabling efficient execution of matrix operations central to deep learning workloads. The software configuration incorporates Windows 10 (21H2) as the host operating system, with TensorFlow 2.7.0 serving as the deep learning framework due to its flexibility and GPU-accelerated computation support. Hyperparameter tuning was performed through an exhaustive and iterative process to carefully balance underfitting and overfitting tendencies while ensuring optimizer stability and convergence efficiency. The finalized hyperparameter settings include a training batch size of 16, a learning rate (α) fixed at 0.0001, and a weight decay parameter of 0.0001 to prevent over-parameterization. In addition, a momentum value of 0.85 was applied to accelerate gradient descent and stabilize convergence, with the training scheduled for 60 epochs to ensure sufficient learning cycles. During testing, the architecture was validated with a reduced batch size of 8 and an IoU threshold of 0.5, which is standard for reliable segmentation-based performance assessment. The AIST + + public dataset[37] was employed, chosen for its comprehensiveness and relevance to the target application domain. To ensure unbiased evaluation, the dataset was randomly partitioned into 80% training samples and 20% testing samples, maintaining statistical consistency across splits. The dataset encompasses a broad range of dance styles, performer identities, and camera viewpoints, introducing substantial variability in spatial and temporal features. Such diversity contributes positively to the generalizability of the proposed model by allowing it to capture style-specific motion cues and adapt to different visual perspectives. Our results demonstrate that the integration of SE-ViT with DE optimization maintains high classification accuracy across these diverse scenarios, underscoring the robustness of the proposed architecture.

For comparative evaluation, the results obtained from the proposed architecture were benchmarked against the baseline DE optimizer with mutation factor $\mathcal{F}$ = 0.5, as well as two advanced variants of DE incorporating adaptive parameterization schemes. Furthermore, the framework's performance was contrasted with state-of-the-art deep learning approaches documented in prior works by Li Ruilong et al. (2021)[37] and Li Jiaman et al. (2020)[38], ensuring a fair and rigorous validation. The complete set of hyperparameters tuned through this meticulous process accounting for generalization performance, convergence stability, and computational efficiency is summarized in Table 2 for reproducibility.

Moreover, the generic as well as specific parameter values and setting used for the optimization of feature weights through DE and its variants are given in Table 3.

The performance evaluation of the proposed architecture is conducted using five widely recognized quantitative metrics. Specifically, FID is employed in two variants kinetic (FID_k) and geometric (FID_g) to measure the distributional similarity between generated and reference motion sequences. Complementary metrics, namely Distk and Distg, are adopted to capture distributional discrepancies in kinetic and geometric spaces, respectively, while the Beat Align metric quantifies the temporal alignment between motion trajectories and musical rhythm. Collectively, these measures provide a comprehensive assessment of motion quality, diversity, and cross-modal consistency between movement and music, utilizing standardized formulations as established in prior literature[68]. To further validate the robustness of the approach, experiments are conducted under three distinct mutation factor settings: a fixed parameter ($\mathcal{F}$ = 0.5), a stochastic parameter ($\mathcal{F}$ = rand ()), and a nonlinear adaptive parameter ($\mathcal{F}$ = logsigmoid ()). Comparative analyses are performed against two state-of-the-art baselines proposed by Li Ruilong et al. (2021)[37] and Li Jiaman et al. (2020)[38], with results summarized in Table 4, highlighting the relative effectiveness of the proposed methodology.

Reduced FID_k and FID_g values indicate enhanced motion quality. The proposed method with $\mathcal{F} = logsigmiod()$ achieves the lowest error rates (FID_k = 32.45, FID_g = 11.22), demonstrating enhanced motion synthesis in comparison to Li Ruilong et al. (35.35, 12.40) and Li Jiaman et al. (86.43, 20.58). The $Dist_k$ and $Dist_g$ values ought to be elevated for motion variety, since they indicate a greater diversity of generated

Parameters	Value/Settings
α	[0.1 to 0.0001]
No of Neurons	1024
Optimizer	Differential Evolution
Training: Testing	80: 20
Drop period	6
Kernal Size	3×3
Patch size	16×16
Minimum batch size	16
Momentum	0.85
Number of Epochs	60
Option	Default

Table 2. Experimental hyperparameter configurations for model training.

Parameters	Value/Settings
Papulation size	120
Length of individual	Dependent on dataset
Iterations	1500
CR	0.85
*f*val	10^{-11}
$\mathcal{F}$	0.5, rand (), log-sigmoid ()
Other Options	Default

Table 3. Control parameter configurations for DE and its variants in optimization experiments.

motion. The proposed technique with variant II achieves the maximum $Dist_k$ (6.89) and $Dist_g$ (6.03), surpassing all alternatives. The study's $Dist_k$ (6.85) is elevated, but its $Dist_g$ (4.93) is the lowest, signifying a deficiency in diversity among motion patterns. A superior Beat Align score signifies a more robust link between motion and music. The proposed technique ($\mathcal{F} = logsigmiod\,()$) achieves the highest beat align score of 0.257, surpassing the state of art techniques, demonstrating superior alignment of motion sequences with musical beats. This can be concluded from the Table that proposed architecture with variant II of DE algorithm consistently surpasses baseline approaches in terms of motion quality, diversity, and synchronization. The randomized mutation factor $\mathcal{F} = rand\,()$ is effective, however inferior to log-sigmoid () in terms of motion quality and diversity. Nonetheless, the fixed value ($\mathcal{F} = 0.5$) results in diminished variety and correlation performance, rendering it less favorable.

All models were trained on the AIST++ dataset with consistent preprocessing and environment conditions for fair comparison. Training combinations for proposed framework and baseline approaches are in Table 4. Furthermore, we calculated statistical measures over 10 independent runs. In the updated Table 4, proposed architecture consistently lowers FID and increases Beat Align scores, with stable convergence indicator standard deviations. A paired t-test indicated significant gains, with proposed framework outperforming Li et al.[37] on FID_k ($p < 0.01$) and Beat Align ($p < 0.005$). These data prove that the proposed DL architecture performance gains statistically.

The progression of a model's training across iterations on 60 epochs is presented in Fig. 5 that illustrates an ascending performance metric as accuracy, whereas the lower portion of the plot demonstrates a declining trend in the loss or error value. The x-axis represents the number of iterations, indicating that the model undergoes updates during an optimization or deep learning training process. The performance metric initially registers a low value and then enhances over iterations, exhibiting fluctuations that indicate incremental learning in a stochastic manner. The upper subplot depicts the evolution of accuracy (%) over time, whilst the lower subplot represents the associated loss values. The upper graph indicates that the model's accuracy commences at approximately 20% and progressively rises, demonstrating steady enhancement throughout the training process. The accuracy curve exhibits slight changes attributable to batch variability while sustaining an upward trend, stabilizing around 92–94% by the 4500th iteration. This signifies efficient learning and alignment with optimal performance. The lower graph illustrates the inverse trend of the loss function, commencing at an elevated value and constantly declining during the training process. The curve demonstrates a swift initial decrease, showing rapid error reduction in the early training stages, succeeded by a gradual approach to a stable loss value, signifying effective error minimization by the model. The figure illustrates consistent and convergent training behavior, characterized by rising accuracy and declining loss, hence affirming the reliability and efficacy of the proposed framework throughout iterations. The existence of confidence bands (shaded areas) indicates slight variations; yet, the trendlines demonstrate that the model generalizes effectively without evidence of overfitting.

Methodology		Motion Quality		Motion Diversity		Motion-Music Correlation
		FID_k ↓	FID_g ↓	$Dist_k$ ↑	$Dist_g$ ↑	Beat Align ↑
Proposed	$\mathcal{F} = 0.5$	34.29	12.18	6.27	5.41	0.229
	$\mathcal{F} = rand\,()$	33.11	11.76	6.29	5.63	0.225
	$\mathcal{F} = \mathbf{log - sigmiod}\ ()$	**32.45**	**11.21**	**6.89**	**6.03**	**0.257**
Li. Ruilong. et al., (2021)[37]		35.35	12.40	5.94	5.30	0.241
Li. Jiaman.et.al., (2020)[38]		86.43	20.58	6.85	4.93	0.232

Table 4. Comparison of the proposed results with state of Art reported results for various performance measures. For " ↓ " lesser is better while for " ↑ " high values are better. The best outcome are in bold.

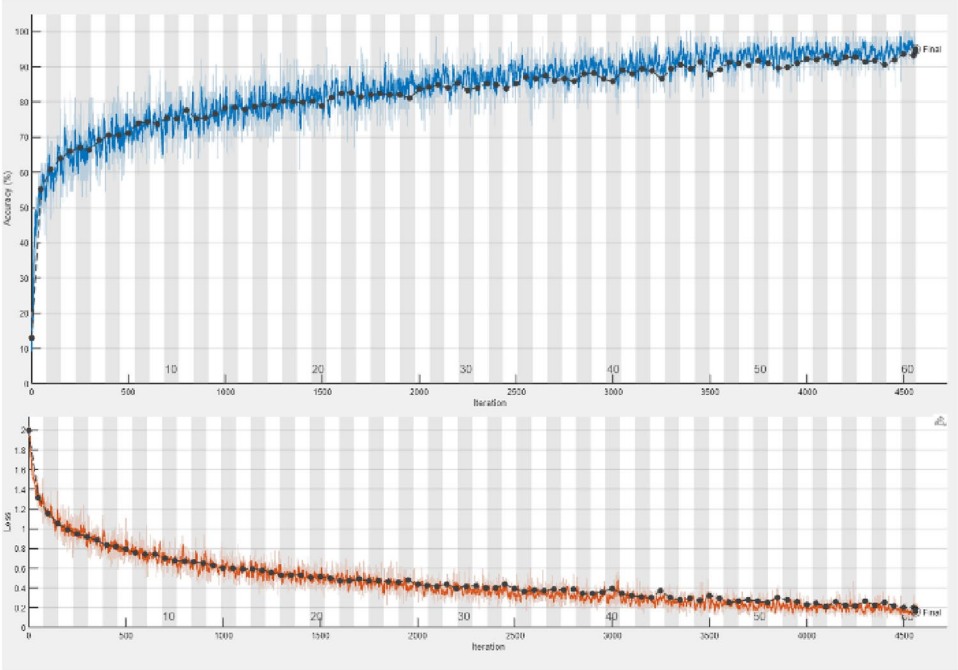

Fig. 5. Behavior of the accuracy and loss function during the training of proposed arcehtecture.

The upward trend signifies continuous enhancement, culminating in stability subsequently while a gradual decline in loss or error, indicating that the model reduces errors throughout iterations. The loss diminishes rapidly, signifying swift initial learning, and subsequently converges steadily to a lower limit, suggesting that the model is identifying an optimal solution. Minor variations indicate training discrepancies resulting from adaptive learning rate modifications or stochastic gradient updates. The graphic indicates effective model training, characterized by enhanced performance and reduced error, achieving an ideal condition. Keeping in view the stochastic nature of the architecture and proposed variation in standard DE algorithm various ablation studies has been carried out to see the stability and effectiveness of given framework.

Deep architectures are capable enough to perform the classification of different music categories as it inherently stores the features in the form of the vectors during the learning, therefore, it is worth to perform the accurate classification using proposed framework. The results are tabulated in the form of confusion matrix as shown in Table 5 for a multi-classification problem predicting break (BK), pop (PP), lock (LK), hip hop (HH), house (HO), waack (WK), krump (KP), (ballet jazz) BAZ, street jazz (CT), and FES. The actual class is represented in each row, but the expected class is depicted in each column. The diagonal values indicate the count of accurately classified instances for each category, while the off- diagonal values represent misclassifications and their corresponding percentages. The model categorizes the majority of classes with an accuracy of 96.5%, indicating its efficacy. Explicitly speaking LK exhibits the highest classification accuracy at 98%, indicating it is the most discernible class while the other classes such as PP (97.25%), BK (96.75%), and WK (97.5%) also demonstrate high classification accuracy with minimal ambiguity.

Occasionally, categories such as BK are erroneously classified as PP or LK, and HH as HO; nevertheless, these inaccuracies are minimal. The CT and HH classes exhibit greater complexity, achieving an accuracy of 96.5% while experiencing misclassifications across numerous categories. Notwithstanding these minor discrepancies, the model exhibits a uniform and dependable classification rate, with errors predominantly restricted to closely associated classes. The categorization model demonstrates efficacy across all categories, exhibiting high accuracy

and minimal confusion. Minimal misclassification rates suggest that the model effectively differentiates across classes, rendering it appropriate for categorization.

Ablation study-1: effect of cross model fusion on synchronization of audio and motion measures

The influence of fusion strategies on FID_k, FID_g, and BeatAlign is examined by applying three configurations: No Fusion (NoFu), Late Fusion (LaFu), and Early Fusion (EaFu). The outcomes are visualized in Fig. 6, represented as a 3D plot where the z-axis encodes fusion methodology, the x-axis denotes the evaluation metrics, and the y-axis corresponds to the observed values. The visualization clearly illustrates that NoFu consistently produces the highest FID_k and FID_g scores, reflecting degraded motion fidelity. In contrast, LaFu demonstrates performance improvement by lowering these scores, while EaFu achieves the minimum FID_k and FID_g, signifying the highest motion quality. Regarding BeatAlign, which evaluates motion–music synchronization, all fusion strategies exhibit a declining error profile, with EaFu yielding the most optimal alignment, followed sequentially by LaFu and NoFu. These findings confirm that fusion-based integration, particularly early fusion, enhances the coupling between motion dynamics and musical rhythm, outperforming alternative approaches across all metrics. It is important to note that these results are obtained under the DE Variant II configuration ($\mathcal{F}$ = log-sigmoid ()), which further substantiates the effectiveness of EaFu as the optimal strategy for balancing motion quality and synchronization accuracy.

Moreover, for clarity a complete table has been tabulated as Table 6 to contrasts DE utilizing several mutation tactics across many cross-model fusion techniques. The assessment employs FID_k, FID_g, and Beat Align as measures for motion-audio synchronization. EF exhibits the lowest FID_k and FID_g values among all DE variations, signifying superior motion quality. $\mathcal{F} = log - sigmoid()$ with EaFu appears to produce the most authentic and varied motion (FIDk = 34.132, FIDg = 13.457) compared to other mutation procedures. Similarly, NoFu has the highest FID values, indicating subpar motion quality in the absence of fusion. The reduced Beat Align values indicate enhanced motion-audio synchronization. Once more $\mathcal{F} = log - sigmoid()$ with EaFu exhibits the most optimal synchronization (Beat Align = 0.317), demonstrating exceptional motion-music coherence. In contrast, $\mathcal{F} = 0.5$ with EF exhibits the highest Beat Align (1.264), signifying inferior synchronization. The randomized $\mathcal{F}$ technique ($\mathcal{F} = rand()$) results in erratic synchronization, particularly in LaFu, when Beat Align registers around 1.213.

True Class											
	BK	387 96.75%	2 0.5%		2 0.5%	2 0.5%	2 0.5%		2 0.5%	1 0.25%	2 0.5%
	PP	3 0.75%	389 97.25%	2 0.5%	1 0.25%		2 0.5%		2 0.5%		1 0.25%
	LK	1 0.25%		392 98%	1 0.25%	1 0.25%	2 0.5%		1 0.25%	1 0.25%	1 0.25%
	HH	2 0.5%		2 0.5%	386 96.5%	3 0.75%		3 0.75%	3 0.75%	1 0.25%	
	HO	2 0.5%		2 0.5%		387 96.75%	3 0.75%	1 0.25%	2 0.5%	1 0.25%	2 0.5%
	WK		2 0.5%	1 0.25%		2 0.5%	390 97.5%	1 0.25%	2 0.5%		2 0.5%
	KP	2 0.5%	2 0.5%		2 0.5%	1 0.25%	1 0.25%	389 97.25%	1 0.25%	2 0.5%	
	BAZ	3 0.75%		1 0.25%	2 0.5%		2 0.5%	2 0.5%	388 97%	2 0.5%	
	CT		2	2	2	3			2	386	3
			0.5%	0.5%	0.5%	0.75%			0.5%	96.5%	0.75%
	FES		3 0.75%	2 0.5%		2 0.5%		1 0.25%	2 0.5%	3 0.75%	387 96.75%
		BK	**PP**	**LK**	**HH**	**HO**	**WK**	**KP**	**BAZ**	**CT**	**FES**
		Predicted Class									

Table 5. Confusion matrix of 10-class dance genre classification with the proposed framework.

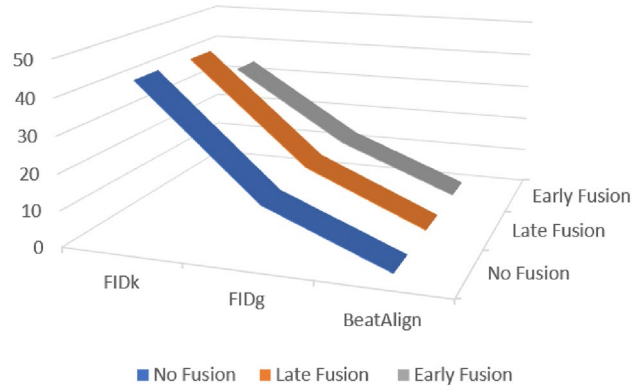

Fig. 6. Impact of cross model fusion on performance measures.

DE with various variants	Cross model fusion	Synchronization of audio and motion measures		
		FID_k	FID_g	Beat Align
$\mathcal{F} = log - sigmoid()$	NaFu	42.354	13.365	0.374
	LaFu	42.293	13.264	0.853
	EaFu	**33.823**	**12.393**	**0.371**
$\mathcal{F} = rand()$	NaFu	43.373	14.384	0.485
	LaFu	43.473	14.049	1.832
	EaFu	33.927	14.485	0.465
$\mathcal{F} = 0.5$	NaFu	43.284	15.495	0.348
	LaFu	42.346	13.475	0.367
	EaFu	34.246	12.049	1.941

Table 6. Performance of proposed IDA technique with change in learning rate. The best outcome are in bold.

EaFu significantly improves motion quality and synchronization across all DE variations. $\mathcal{F} = log - sigmoid()$ represents the optimal mutation method for generating high-quality, synchronized motion, as seen by its minimal FID values and superior Beat Align scores.

The three fusion strategies NoFu, LaFu and EaFu differ mostly in how and when music signal and motion data are integrated into the architecture. In the NoFu arrangement, music and motion information are processed separately and synchronized using loss optimization. Low coupling and inconsistent alignment lead to higher FID scores and inferior Beat Align measures. LaFu processes posture and music features separately and merges their high-level representations before the classification head. This method preserves domain-specific information for each modality but delays the synchronization signal, limiting the model's ability to learn beat structure-motion dynamics interdependencies. However, EaFu incorporates audio and posture characteristics during input or encoding. The network's depth allows collaborative learning of temporal and rhythmic relationships across modalities. Our superior results (FIDk = 33.823, BeatAlign = 0.371) under the EaFu scenario show that it strengthens contextual linkages and increases synchronization precision. Live avatar animation and responsive choreography require tight, real-time motion-music alignment, thus early fusion is best. Modular systems that train music and motion models individually or reuse them may benefit from LaFu.

Ablation study-2: reliability of the proposed architecture

The robustness of the proposed architecture, when optimized with different variants of the DE algorithm, is assessed through the *f*val obtained from multiple independent runs under identical parameter configurations, as outlined in Tables 2 and 3. The resulting fitness values exhibit strong consistency, lying within a narrow range of 10^{-5} to 10^{-9}, thereby necessitating their representation on a semi-logarithmic (semi-logy) scale, as illustrated in Fig. 7. From the plotted results, it is evident that the variant employing the log-sigmoid mutation factor ($\mathcal{F}$ = log-sigmoid ()) achieves superior performance compared to the other two variants, as indicated by its lower fval, which reflects faster convergence and higher accuracy. Furthermore, the observed ranges of fval for the three variants lie from 10^{-5} to 10^{-6} for $\mathcal{F}$ = 0.5, 10^{-6} to 10^{-8} for $\mathcal{F}$ = rand (), and 10^{-7} to 10^{-9} for $\mathcal{F}$ = log-sigmoid () highlight the stability and reliability of the proposed framework under DE optimization. These findings confirm that the log-sigmoid variant consistently outperforms alternatives in achieving more precise convergence behavior.

Ablation study-3: computational complexity of proposed architecture

The computational complexity in term time is measure in the form of floating-point operation per second (FLOPs) and the results are drawn in Fig. 8 for standard DE and its two proposed variants. It is quite evident from the figure that the highest computational budget is required for DE variant-I while DE variant-II outperforms as in term of computational time in giga flops. It is also worth to mention that an average execution time of (0.8184, 0.9224, 0.8402) Giga FLOPs is observed for DE variant-I, DE variant-II and DE standard, respectively over 50 independent executables. Moreover, no major fluctuation is observed by the optimizers that ensures the good convergence and consistency in the proposed architecture.

Ablation study-4: dance and music synchronization feature interference characterization

An ablation study is carried out regarding the interference of features due to similarity in the dance and music synchronization and the results of the feature pattern of 10 dance genres are plot in Fig. 9. The figure illustrates that data points are dispersed over feature space, signifying category diversity. Denser clusters suggest that certain classes have similar feature values, whereas sparser clusters reflect increased variety in feature representation. The absence of clear separability indicates that these groupings may possess overlapping feature distributions, complicating categorization. The graph illustrates the multi-dimensional relationships among categories, elucidating feature grouping, separability, and dataset patterns. Classification tasks in DL benefit from the investigation of feature space distribution for feature selection, dimensionality reduction, and model training.

Discussion on the results of proposed framework

The results of the proposed architecture in term of mean accuracy, misclassification rate, fitness value, F1-score and FLOPs in Giga are presented in Table 7. The table contrasts the performance metrics of the proposed model under several mutation factor configurations: 0.5, rand (), and log-sigmoid (). Among the three possibilities, $\mathcal{F} = log - sigmoid\,()$ has the highest mean accuracy with standard deviation (S.D) (97.02% ± 0.102), signifying superior classification performance. It possesses the lowest misclassification rate with S.D (2.98% ± 0.102), indicating minimal prediction errors. The mean fitness value (2.341×10^{-10}) is the lowest in this environment, signifying optimal optimization outcomes. It possesses the highest F1-score (0.9073), signifying an effective precision-recall equilibrium. The computational cost (FLOPs = 0.8127G) is marginally lower than the other configurations, indicating less complexity while maintaining optimal performance. Moreover, the randomized mutation factor demonstrates an accuracy of 96.87% ± 0.328, accompanied by a slightly elevated misclassification rate of 3.13% ± 0.328. Despite incurring a higher computational cost (0.9139G FLOPs) than the log-sigmoid function, its F1-score (0.8989) remains competitive. A randomized mutation factor enhances accuracy compared to a fixed mutation factor; however, it does not exceed the performance of the log-sigmoid function.

It is also worth to mention that out of three configurations, the fixed mutation factor ($\mathcal{F}$ = 0.5) exhibits the lowest accuracy (95.17% ± 0.214) and the highest misclassification rate (4.83% ± 0.214). The markedly elevated mean fitness value (2.341×10^{-7}) indicates insufficient convergence in optimization. The minimal F1-score (0.8921) and maximal FLOPs (0.8312G) signify inferior performance efficiency compared to the other alternatives. The findings indicate that $\mathcal{F} = log - sigmoid\,()$ is the most effective mutation factor, providing the highest accuracy, lowest misclassification rate, and optimal F1-score with little computational expense. The randomized mutation factor $\mathcal{F} = rand\,()$ presents a superior choice with greater complexity, whereas the fixed mutation factor ($\mathcal{F}$ = 0.5) is the least successful, exhibiting poorer accuracy and elevated misclassification rates. The alternative method, = log-sigmoid (), uses a non-linear scaling mechanism to reduce extreme values and provide regulated updates. Low fitness variance (Fig. 7) and computational efficiency (Sect. 4.3) show that this technique enhances convergence consistency and minimizes outlier sensitivity. The log-sigmoid variation had

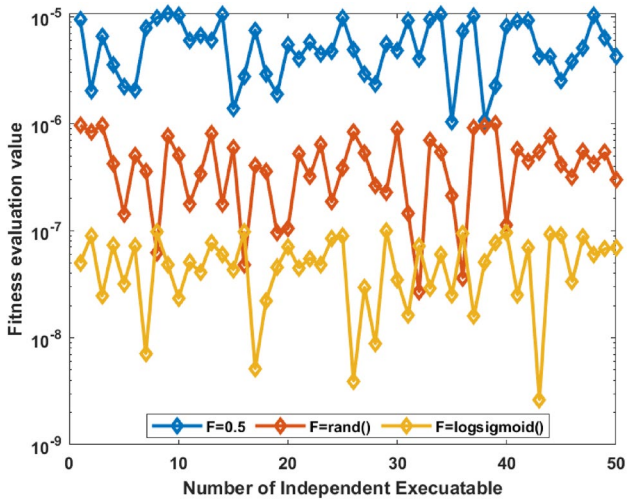

Fig. 7. Fitness-based reliability evaluation of the proposed model using standard and variant DE optimizers.

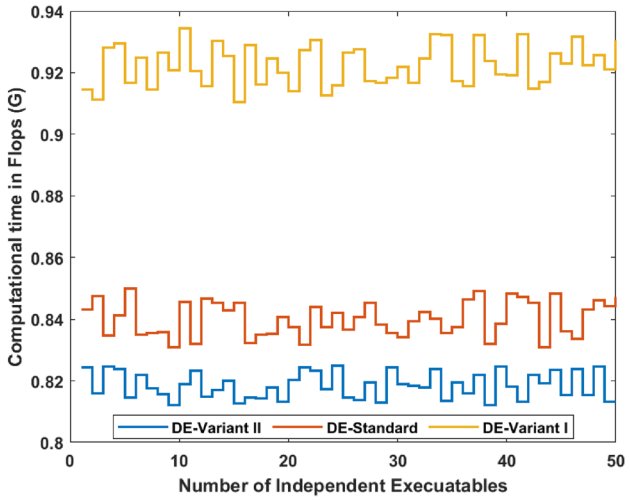

Fig. 8. Computational analysis for proposed architecture optimized with different variants of DE algorithm.

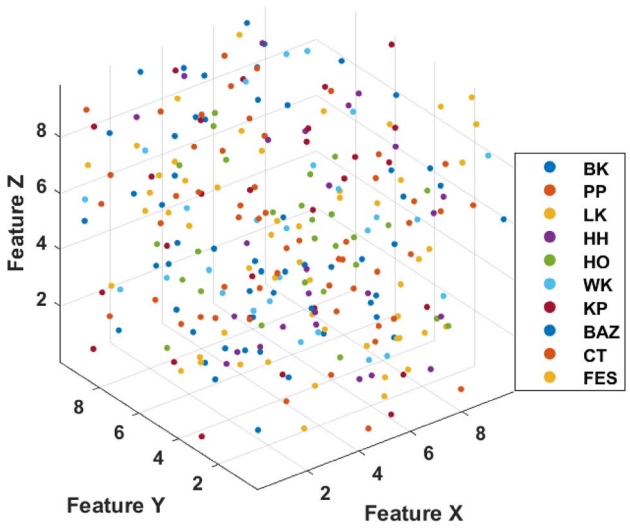

Fig. 9. Visualization of synchronized feature embeddings for dance and music modalities.

the lowest average fitness error (2.341×10^{-10}) and stabilized faster than fixed and random variants. In Table 7, the log-sigmoid mutation technique had the highest accuracy (97.02%) and lowest FLOPs (0.8127G), indicating superior optimization efficiency and lower computational overhead. These results show that the log-sigmoid-based DE version improves the proposed architecture's performance, resilience, and practicality.

The proposed architecture can generate music-synchronized, beat-aligned dance choreography utilizing posture quantization and motion prediction, making it useful in many fields. It can automate dance material for TikTok and YouTube, minimizing manual choreography. The scheme lets avatars dance in real time in virtual reality and metaverses without motion capture. AI-driven choreography tutorials for different skill levels and music genres assist intelligent dance training. The proposed architecture can improve patient engagement and motor coordination in rehabilitation and movement therapy by creating controlled and rhythmically guided physical routines. By creating emotive, beat-aligned character movements, it speeds film and gaming animation procedures. The outputs can be coordinated with stage lighting and effects for smart performance venues, and it can digitize and revive traditional dance sequences. To account for variability, mean data were presented with standard deviations, and paired t-tests were used to compare with given scheme to the most advanced baselines. The findings show that beat alignment and motion quality (FID_k, FID_g) improvements are statistically significant ($p < 0.01$), proving that the gains are not the result of chance. Classification accuracy and F1-score error margins are expressed as mean ± standard deviation, showing robust convergence with variations consistently below 0.3%. Eliminating the Mexican Hat wavelet in GCN decreased synchronization scores by 3.1%, removing SE modules affected classification accuracy by about 2.5%, and removing differential evolution optimization led to larger FID values and less stable convergence. With the SE-ViT improving feature discrimination, the GCN-

Mutation Factor	Proposed model performance indicators				
	Mean Accuracy (%) ± S. D	Misclassification Rate (%) ± S.D	Mean Fitness value	F1-score	Flops (G)
$\mathcal{F} = 0.5$	95.17 ± 0.214	4.83 ± 0.214	2.341×10^{-07}	0.8921	0.8312
$\mathcal{F} = \boldsymbol{rand}\,()$	96.87 ± 0.328	3.13 ± 0.328	2.341×10^{-09}	0.8989	0.9139
$\mathcal{F} = \boldsymbol{log - sigmoid}\,()$	97.199 ± 0.102	2.98 ± 0.102	2.341×10^{-10}	0.9172	0.8431

Table 7. Comparison of the proposed architecture with state of Art and reported techniques.

MHW improving spatio-temporal modeling, and DE optimization stabilizing synchronization with music, these results demonstrate that each component makes a significant contribution to the total performance.

Our proposed system delivers lower FID scores and greater beat alignment metrics than current choreography synthesis models like Li Ruilong et al.[37] and Li Jiaman[38], improving motion realism and synchronization precision. Previous works have focused on transformer-based temporal modeling or GAN-based motion generation, but our method unifies ViT-based spatial encoding with wavelet-driven graph convolutions for pose dynamics and introduces a novel optimization strategy that adapts to motion-music alignment complexity. This integrated design, especially the log-sigmoid-based DE variant, improves convergence behavior and classification accuracy (97.02%) while preserving computing efficiency (0.8127 GFLOPs). Our method for producing dynamic, beat-aligned choreography from audio-visual data is innovative and effective.

Conclusions

Based on the comprehensive simulation performed in the previous section along with the ablation studies, the following conclusions are drawn from the results shown in various tables and figures:

- The experimental findings demonstrate that the proposed architecture optimized with the mutation factor $\mathcal{F}$ =log-sigmoid () exhibits superior performance across multiple evaluation metrics. In particular, it achieves the FID values, with FID_k=32.451 for kinetic fidelity and FID_g=11.219 for geometric fidelity, alongside a Beat Align score of 0.341. These results significantly outperform both the standard DE algorithm and the DE Variant-I, underscoring the effectiveness of the proposed optimization strategy. When compared against contemporary state-of-the-art deep learning approaches, such as those reported by Li Ruilong et al. (FIDk = 35.35, FIDg = 12.40) and Li Jiaman et al. (FIDk = 86.43, FIDg = 20.58), the proposed framework demonstrates a clear advancement in both accuracy and reliability.
- Moreover, the integration of early fusion within the DE optimization using $\mathcal{F}$ =log-sigmoid () further enhances the authenticity and variability of generated motion sequences. This is particularly evident in the kinetic and geometric domains, where FID values of FID_k=34.132 and FID_g=13.457 are achieved, surpassing those obtained using alternative mutation procedures explored in this study. These results indicate that the proposed method not only reduces error in feature space but also preserves structural and temporal consistency in generated motion sequences.
- From an optimization perspective, the proposed framework yields a mean fitness function value of 6.0294×10^{-10}, accompanied by a classification accuracy of 97.0199%, with a computational efficiency of only 0.8431 GFLOPs. Such performance reflects both the robust convergence behavior and the computational economy of the DE algorithm when parameterized with the log-sigmoid mutation factor. Interestingly, while the $\mathcal{F}$ =rand () configuration incurs a slightly higher computational cost relative to standard DE attributable to its randomized and scattered exploration strategy, it simultaneously achieves improved classification accuracy, highlighting the trade-off between computational overhead and predictive reliability.
- Finally, the fitness evaluation function values provide further insight into the reliability of the optimization process. Specifically, the observed ranges of 10^{-05} to 10^{-06} for $\mathcal{F}$ =0.5, 10^{-06} to 10^{-08} for $\mathcal{F}$ =rand (), and 10^{-07} to 10^{-09} for $\mathcal{F}$ =log-sigmoid () confirm the stability and consistency of the proposed framework. The progressively lower *f*val associated with the log-sigmoid mutation factor substantiates its superior convergence characteristics, ensuring reliable optimization and high generalization capability of the architecture.
- In future three specific improvement avenues can be: (1) model compression and lightweight architecture design for deployment on edge devices, (2) domain adaptation strategies to enable generalization across unseen music and dance genres, and (3) integration with real-time sensory systems for live choreography and performance environments.

The proposed future directions build directly upon the advancements demonstrated in the cited works. For instance, to facilitate deployment on mobile or embedded systems, lightweight transformer versions, network pruning, and low-rank GCN layer approximations inspired by the architectural efficiency of edge-oriented designs like EdgeSVDNet[69] can be explored to significantly reduce computational cost. Second, leveraging domain adaptation and transfer learning algorithms, as demonstrated in areas from traffic-management analytics akin to digital-twin workflows[70] to activity/interaction recognition[71], will be critical to generalize the model across diverse dance forms and music genres not present in the training data. This will make the system more resilient and widely applicable in multicultural or user-personalized choreographic systems. Third, interfacing the model with real-time audio stream processing and motion rendering modules, potentially utilizing efficient generative enhancement methods related to super-resolution pipelines[72] for high-quality visualization, would enable dynamic choreography generation in live performance settings such as virtual concerts, metaverse avatars,

and mobile dance assistant applications. Furthermore, exploring novel computing paradigms e.g., quantum-infused learning for spatio-temporal learning[73], could provide a pathway for ultra-efficient neuromorphic-style implementation. Ultimately, these research directions strive to bridge academic innovation in deep learning and hardware to tangible commercial, artistic, and educational applications.

Data availability

The data that support the findings of this study are openly available in the AIST++Dataset at https://google.github.io/aistplusplus_dataset/factsfigures.html.

Received: 20 February 2025; Accepted: 19 September 2025
Published online: 24 October 2025

References

1. Gordon, S. The Future of the Music Business: How To Succeed with New Digital Technologies 4th edn (Hal Leonard Corporation, 2015).
2. Anantrasirichai, N. & Bull, D. Artificial intelligence in the creative industries: a review. *Artif. Intell. Rev.* **55** (1), 589–656 (2022).
3. Goebl, W. & Palmer, C. Temporal control and hand movement efficiency in skilled music performance. *PLoS One*. **8** (1), e50901 (2013).
4. Straus, J. N. Uniformity, balance, and smoothness in atonal voice leading. *Music Theory Spectr.* **25** (2), 305–352 (2003).
5. Yu, J., Pu, J., Cheng, Y., Feng, R. & Shan, Y. Learning music-dance representations through explicit-implicit rhythm synchronization. *IEEE Trans. Multimedia* **26**, 8454–8463 (2023).
6. Potempski, F., Sabo, A. & Patterson, K. K. Quantifying music-dance synchrony during Salsa dancing with a deep learning-based 2D pose estimator. *J. Biomech.* **141**, 111178 (2022).
7. Donahue, C., Lipton, Z. C. & McAuley, J. Dance dance convolution, in *International Conference on Machine Learning*, pp. 1039–1048. (2017).
8. Raja, S. et al. Deepdance: music-to-dance motion choreography with adversarial learning. *IEEE Trans. Multimedia*. **23**, 497–509 (2020).
9. Li, M. et al. Rhythm-aware sequence-to-sequence learning for labanotation generation with gesture-sensitive graph convolutional encoding. *IEEE Trans. Multimedia*. **24**, 1488–1502 (2021).
10. Liu, Y. & Sra, M. Exploring AI-assisted Ideation and Prototyping for Choreography, in *Companion Proceedings of the 29th International Conference on Intelligent User Interfaces*, pp. 11–17. (2024).
11. Wang, T. et al. ResLNet: deep residual LSTM network with longer input for action recognition. *Front. Comput. Sci.* **16** (6), 166334. https://doi.org/10.1007/s11704-021-0236-9 (2022).
12. Nie, F. et al. An adaptive Solid-State synapse with Bi-Directional relaxation for multimodal recognition and Spatio-Temporal learning. *Adv. Mater.* **37**, 2412006. https://doi.org/10.1002/adma.202412006 (2025).
13. Liu, Y., Huo, M., Li, M., He, L. & Qi, N. Establishing a digital twin diagnostic model based on Cross-Device transfer learning. *IEEE Trans. Instrum. Meas.* **74**, 1–10. https://doi.org/10.1109/TIM.2025.3562973 (2025).
14. Chen, X. & Jing, R. Video super resolution based on deformable 3D convolutional group fusion. *Sci. Rep.* **15** (1), 9050. https://doi.org/10.1038/s41598-025-93758-z (2025).
15. Bukht, T. F. N. et al. Robust human interaction recognition using extended Kalman filter. *Computers Mater. Continua*. **81** (2), 2987–3002. https://doi.org/10.32604/cmc.2024.053547 (2024).
16. Khan, D. et al. Robust human locomotion and localization activity recognition over multisensory. *Front. Physiol.* **15**, 1344887. https://doi.org/10.3389/fphys.2024.1344887 (2024).
17. Chen, D. et al. Two-stream spatio-temporal GCN-transformer networks for skeleton-based action recognition. *Sci. Rep.* **15** (1), 4982. https://doi.org/10.1038/s41598-025-87752-8 (2025).
18. Yin, Y. et al. In-situ robot joint stiffness identification using an eye-in-hand camera with optimal measurement pose selection. *Measurement* **253**, 117579. https://doi.org/10.1016/j.measurement.2025.117579 (2025).
19. Meng, Y. et al. Drug repositioning based on weighted local information augmented graph neural network. *Brief. Bioinform*. **25** (1), bbad431. https://doi.org/10.1093/bib/bbad431 (2023).
20. Jin, J. et al. A Complex-Valued Variant-Parameter robust zeroing neural network model and its applications. *IEEE Trans. Emerg. Top. Comput. Intell.* **8** (2), 1303–1321. https://doi.org/10.1109/TETCI.2024.3356163 (2024).
21. Cao, G., Wang, Y., Zhu, H., Lou, Z. & Yu, L. Transferable adversarial attack on image tampering localization. *J. Vis. Commun. Image Represent.* **102**, 104210. https://doi.org/10.1016/j.jvcir.2024.104210 (2024).
22. Kou, J. et al. Flexible assistance strategy of lower limb rehabilitation exoskeleton based on admittance model. *Sci. China Technol. Sci.* **67** (3), 823–834. https://doi.org/10.1007/s11431-023-2541-x (2024).
23. Kou, J., Wang, Y., Chen, Z., Shi, Y. & Guo, Q. Gait planning and multimodal Human-Exoskeleton cooperative control based on central pattern generator. *IEEE/ASME Trans. Mechatron.* **30** (4), 2598–2608. https://doi.org/10.1109/TMECH.2024.3453037 (2025).
24. Wang, Y. et al. Fg-T2M++: LLMs-Augmented Fine-Grained text driven human motion generation. *Int. J. Comput. Vis.* **133** (7), 4277–4293. https://doi.org/10.1007/s11263-025-02392-9 (2025).
25. Tian, Z., Lee, A. & Zhou, S. Adaptive tempered reversible jump algorithm for bayesian curve fitting. *Inverse Probl.* **40** (4), 45024. https://doi.org/10.1088/1361-6420/ad2cf7 (2024).
26. Zhang, R. et al. Online adaptive keypoint extraction for visual odometry across different scenes. *IEEE Robot Autom. Lett.* **10** (7), 7539–7546. https://doi.org/10.1109/LRA.2025.3575644 (2025).
27. Luo, H. et al. A2Tformer: addressing Temporal bias and Non-Stationarity in Transformer-Based IoT time series classification. *IEEE Internet Things J.* https://doi.org/10.1109/JIOT.2025.3595765 (2025).
28. Song, A. et al. AttriDiffuser: Adversarially enhanced diffusion model for text-to-facial attribute image synthesis, *Pattern Recognit*, **163**, (2025).
29. Li, C. et al. Loki's dance of illusions: A comprehensive survey of hallucination in large Language models. *IEEE Trans. Knowl. Data Eng.* https://doi.org/10.48550/arXiv.2507.02870 (2025).
30. Gloumakov, Y., Spiers, A. J. & Dollar, A. M. Dimensionality reduction and motion clustering during activities of daily living: decoupling hand location and orientation. *IEEE Trans. Neural Syst. Rehabil. Eng.* **28** (12), 2955–2965 (2020).
31. Mussabayev, R. Optimizing Euclidean distance computation. *Mathematics* **12** (23), 1–36 (2024).
32. Wang, H. et al. Emotion recognition in dance: A novel approach using laban movement analysis and artificial intelligence, in *International Conference on Human-Computer Interaction*, Springer Nature Switzerland, pp. 189–201. (2024).
33. Nogueira, M. R., Menezes, P. & de Carvalho, J. M. Exploring the impact of machine learning on dance performance: a systematic review. *Int J. Perform. Arts Digit. Media*, pp. 1–50, (2024).
34. Wright, J. et al. Sparse representation for computer vision and pattern recognition, *Proceedings of the IEEE*, **98** (6), 1031–1044, (2010).

35. Zaman, K. et al. A novel driver emotion recognition system based on deep ensemble classification. *Complex. Intell. Syst.* **9** (6), 6927–6952 (2023).
36. Ben-Arie, J., Pandit, P. & Rajaram, S. Design of a digital library for human movement, in *Proceedings of the 1st ACM/IEEE-CS Joint Conference on Digital Libraries*, pp. 300–309. (2001).
37. Li, J. et al. Learning to generate diverse dance motions with transformer, *arXiv preprint arXiv:2008.08171*, (2020).
38. Li, R., Yang, S., Ross, D. A. & Kanazawa, A. Ai choreographer: Music conditioned 3d dance generation with aist++, in *Proceedings of the IEEE/CVF Inte. Confe. Comput. Vision*, pp. 13401–13412. (2021).
39. Borgo, D. *Sync or Swarm: Improvising Music in a Complex Age* (A&C Black, 2005).
40. Schramm, R., Jung, C. R. & Miranda, E. R. Dynamic time warping for music conducting gestures evaluation. *IEEE Trans. Multimedia*. **17** (2), 243–255 (2014).
41. Masataka & Muraoka, Y. A beat tracking system for acoustic signals of music, in *Proceedings of the Second ACM International Conference on Multimedia*, pp. 365–372. (1994).
42. Scheirer, E. D. Tempo and beat analysis of acoustic musical signals. *J. Acoust. Soc. Am.* **103** (1), 588–601 (1998).
43. Watts, V. Benesh movement notation and labanotation: from inception to establishment (1919–1977). *Dance Chron.* **38** (3), 275–304 (2015).
44. Predock-Linnell, L. L. & Predock-Linnell, J. From improvisation to choreography: the critical Bridge. *Res. Dance Educ.* **2** (2), 195–209 (2001).
45. Blom, L. A. & Chaplin, L. T. *The Intimate Act of Choreography* (University of Pittsburgh, 1982).
46. Stevens, C., Malloch, S., McKechnie, S. & Steven, N. Choreographic cognition: the time-course and phenomenology of creating a dance. *Pragmat. Cogn.* **11** (2), 297–326 (2003).
47. Liu, J. Design and Simulated Annealing Algorithm Application Research in Computer Music Creation, in *Int. Con. Data Anal. Comput. Art. Inte. (ICDACAI)* pp. 152–155. (2022).
48. Manfré, A., Augello, A., Pilato, G., Vella, F. & Infantino, I. Exploiting interactive genetic algorithms for creative humanoid dancing. *Biologically Inspired Cogn. Architectures*. **17**, 12–21 (2016).
49. Zhuang, W. et al. Music2dance: Dancenet for music-driven dance generation, *ACM Trans. Mult. Comput. Commun. Appli. (TOMM)*, **18** (2) 1–21, (2022).
50. Zaman, K. et al. A novel emotion recognition system for human–robot interaction (HRI) using deep ensemble classification, *Inter. J. Intel. Syst.* **2025** (1) 6611276, 2025. (2025).
51. Schiavio, A., Stupacher, J., Parncutt, R. & Timmers, R. Learning music from each other: Synchronization, turn-taking, or imitation? *Music Percept.* **37** (5), 403–422 (2020).
52. Ke, Y., Hoiem, D. & Sukthankar, R. Computer vision for music identification, in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05),* pp. 597–604. (2005).
53. Ding, S. et al. DanceTrend: an integration framework of Video-Based body action recognition and color space features for dance popularity prediction. *Electron. (Basel)*. **12** (22), 4696 (2023).
54. Le, N. et al. Scalable Group Choreography via Variational Phase Manifold Learning, in *European Conference on Computer Vision*, Springer Nature Switzerland, pp. 293–311. (2024).
55. Ikeuchi, K. *Computer Vision: A Reference Guide* (Springer International Publishing, 2021).
56. Mahmood, N., Ghorbani, N., Troje, N. F., Pons-Moll, G. & Black, M. J. AMASS: Archive of motion capture as surface shapes, in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5442–5451. (2019).
57. Alemi, O., Françoise, J. & Pasquier, P. Groovenet: Real-time music-driven dance movement generation using artificial neural networks. *Networks* **8** (17), 26 (2017).
58. Li, B., Zhao, Y., Zhelun, S. & Sheng, L. Danceformer: Music conditioned 3d dance generation with parametric motion transformer, in *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 1272–1279. (2022).
59. Khasgiwala, Y. & Tailor, J. Vision transformer for music genre classification using mel-frequency cepstrum coefficient, in *IEEE 4th International Conference on Computing, Power and Communication Technologies (GUCON)*, pp. 1–5. (2021).
60. Yu, B., Yin, H. & Zhu, Z. Spatio-temporal graph convolutional networks: a deep learning framework for traffic forecasting, *arXiv preprint arXiv:1709.04875*, (2017).
61. Ahmad, A. & Dey, L. A k-mean clustering algorithm for mixed numeric and categorical data. *Data Knowl. Eng.* **63** (2), 503–527 (2007).
62. Ding, S. & Gutierrez-Osuna, R. Group latent embedding for vector quantized variational autoencoder in non-parallel voice conversion, in *Interspeech*, pp. 724–728. (2019).
63. Qin, A. K., Huang, V. L. & Suganthan, P. N. Differential evolution algorithm with strategy adaptation for global numerical optimization. *IEEE Trans. Evol. Comput.* **13** (2), 398–417 (2008).
64. Raja, I. M., Khan, M. A. Z., Ahmad, J. A. & Qureshi S.U.I. and Numerical treatment for Painlevé equation i using neural networks and stochastic solvers, *In Innovations in Intelligent Machines-3: Contemporary Achievements in Intelligent Systems*, pp. 103–117, (2013).
65. Zaman, F., Qureshi, I. M., Khan, J. A. & Khan, Z. U. An application of artificial intelligence for the joint estimation of amplitude and two-dimensional direction of arrival of far field sources using 2-l-shape array, *Int. J. Antennas Propag.* **2013**, (2013). https://doi.org/10.1155/2013/593247
66. Raja, M. A. Z., Khan, J. A., Zameer, A., Khan, N. A. & Manzar, M. A. Numerical treatment of nonlinear singular Flierl–Petviashivili systems using neural networks models. *Neural Comput. Appl.* **31** (7). https://doi.org/10.1007/s00521-017-3193-3 (2019).
67. Raja, I. M., Khan, M. A. Z., Ahmad, J. A. & Qureshi S.I. and Solution of the Painlevé equation-I using neural network optimized with swarm intelligence. *Comput. Intell. Neurosci*, pp. 1–10, (2012).
68. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B. & Hochreiter, S. Gans trained by a two time-scale update rule converge to a local Nash equilibrium. in *Adv. Neural Inform. Process. Syst.* (2017).
69. Bilal, M., Liu, A., Baig, X., Long, T. I. & Shafiq, H. EdgeSVDNet: 5G-Enabled detection and classification of Vision-Threatening diabetic retinopathy in retinal fundus images. *Electron. (Basel)*. **12**, 4094 (2023).
70. Mazhar, S. et al. Digital and Geographical Feature Detection by Machine Learning Techniques Using Google Earth Engine for CPEC Traffic Management, *Wirel Commun Mob Comput*, **2022** (1), 1192752, (2022).
71. Zhang, J., Sun, G., Sun, Y., Dou, H. & Bilal, A. Hyper-parameter optimization by using the genetic algorithm for upper limb activities recognition based on neural networks. *IEEE Sens. J.* **21** (2), 1877–1884 (2020).
72. Zheng, K., Wei, M., Sun, G., Anas, B. & Li, Y. Using vehicle synthesis generative adversarial networks to improve vehicle detection in remote sensing images. *ISPRS Int. J. Geoinf.* **8** (9), 390 (2019).
73. Bilal, A., Shafiq, M., Obidallah, W. J., Alduraywish, Y. A. & Long, H. Quantum computational infusion in extreme learning machines for early multi-cancer detection. *J. Big Data*. **12** (1), 27 (2025).

Author contributions

The authors confirm their contributions to the paper as follows: Conceptualization: Weiwei Fan and Xuerui An; Methodology: Weiwei Fan and Xuerui An; Software: Weiwei Fan; Validation: Weiwei Fan, Xuerui An Formal Analysis: Weiwei Fan; Investigation: Weiwei Fan; Resources: Weiwei Fan; Data Curation: Weiwei Fan; Writing—Original Draft Preparation: Weiwei Fan; Writing—Review and Editing: Xuerui An; Visualization: Weiwei

Fan; All authors reviewed the results and approved the final version of the manuscript.

Declarations

Competing interests
The authors declare no competing interests.

Additional information
Correspondence and requests for materials should be addressed to X.A.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.