AI POWERED CHOREOGRAPHY

A PROJECT REPORT

Submitted by

ABDULLA FASEEH ALI M (MES21AD002)
MUHAMMED MUSTHAFA (MES21AD044)
SHAAZ ZABRIN P (MES21AD053)
VAISAKH P K (MES21AD059)

to

The APJ Abdul Kalam Technological University

in partial fulfillment of the requirements for the award of the Degree

of

Bachelor of Technology

in

Artificial Intelligence and Data Science

Department of Artificial Intelligence and Data Science
[B.Tech. Programme]
MES College of Engineering Kuttippuram
Thrikkanapuram South P.O, Malappuram Dt, Kerala, India 679582

2024 - 2025

DECLARATION

We hereby declare that the project report "AI Powered Choreography", submitted for partial fulfillment of the requirements for the award of degree of Bachelor of Technology of the APJ Abdul Kalam Technological University, Kerala is a bonafide work done under the supervision of Ms. Bhavya Parvathi P, Assistant Professor, Dept. of Artificial Intelligence and Data Science. This submission represents our ideas in our own words and where ideas or words of others have been included, We have adequately and accurately cited and referenced the original sources. We also declare that we have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in our submission. We understand that any violation of the above will be a cause for disciplinary action by the institute and the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree, diploma or similar title of any other University.

Place: Kuttippuram
Date: 27/03/2025

ABDULLA FASEEH ALI M

MUHAMMED MUSTHAFA

SHAAZ ZABRIN P

VAISAKH P K

DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

MES COLLEGE OF ENGINEERING, KUTTIPPURAM

CERTIFICATE

This is to certify that the report entitled **"AI POWERED CHOREOGRAPHY"** submitted by **MUHAMMED MUSTHAFA, SHAAZ ZABRIN P, VAISAKH P K, ABDULLA FASEEH ALI M** to the APJ Abdul Kalam Technological University in partial fulfillment of the requirements for the award of the Degree of Bachelor of Technology in Artificial Intelligence and Data Science is a bonafide record of the project work carried out under our guidance and supervision. This report in any form has not been submitted to any other University or Institute for any purpose.

Internal Supervisor:
Ms. Bhavya Parvathi P
Assistant Professor
Dept of ADS

Project Coordinator:
Mr. Sherikh K K
Assistant Professor
Dept of ADS

Head of Department:
Dr. Govindaraj E
Associate Professor
Dept of ADS

ACKNOWLEDGEMENT

At the outset, we would like to thank the Almighty for all his blesssings that led us here.

We are grateful to *Dr. Rahumathunza I, Principal, MES College of Engineering, Kuttippuram*, for providing the right ambiance to complete this project. We would also like to extend our sincere gratitude to *Dr. Govindaraj E, Head of Department, Artificial Intelligence and Data Science, MES College of Engineering, Kuttippuram.*

We are deeply indebted to the project coordinator, *Mr. Sherikh K K, Assistant Professor, Department of Artificial Intelligence and Data Science* for his continued support.

It is with great pleasure that we express our deep sense of gratitude to our project guide, *Ms. Bhavya Parvathi P, Assistant Professor, Department of Artificial Intelligence and Data Science*, for her guidance, supervision, encouragement and valuable advice in each and every phase.

We would also like to express our sincere thanks and gratitude to all staff members of the department, our friends and family members for their cooperation, positive criticism, consistent support and consideration during the preparation of this work.

ABDULLA FASEEH ALI M

MUHAMMED MUSTHAFA

SHAAZ ZABRIN P

VAISAKH P K

ABSTRACT

Choreographing dance sequences manually requires significant effort and expertise. This work presents an AI-based approach to generate synchronized dance movements from music input. The system extracts musical features such as tempo, rhythm, and beat and maps them to motion sequences using a GAN with an LSTM-based generator and discriminator. The generated motion data is then animated using SMPL-X, ensuring realistic movement representation. Dataset preprocessing, model training, and evaluation are conducted using the FineDance dataset. The results demonstrate that AI-generated dance sequences exhibit a reasonable correlation with the input music, contributing to automated choreography.

CONTENTS

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

Choreographing dance movements manually is a complex and time-consuming process that requires expertise in music and movement coordination. Traditional choreography relies on human choreographers to design dance routines, making it a labor-intensive and subjective task. With the increasing use of artificial intelligence in creative fields, there is a growing interest in automating dance generation based on music input. The FineDance dataset, which contains paired music and motion sequences across multiple dance genres, provides a valuable resource for training AI models in this domain[1].

This work presents an AI-driven approach to generate dance movements from music. The system processes musical features such as tempo, rhythm, and beat and maps them to dance sequences using a GAN with an LSTM-based generator and discriminator[8][9]. The generator learns to produce natural dance movements that align with the extracted music features, while the discriminator ensures the generated dance follows realistic motion patterns. The generated motion sequences are then processed and visualized using SMPL-X, creating expressive and fluid dance representations[10][11].

The dataset used for training and evaluation is FineDance, which contains paired music and motion sequences across multiple dance genres. The system is trained to understand different dance styles and their relationships with musical patterns, allowing it to create varied dance movements based on input music. The training process ensures that the model captures the temporal structure of dance, helping to produce smooth and expressive motions.

To enhance motion quality, several preprocessing techniques are applied, including data segmentation, normalization, and feature extraction. The music and motion data are divided into structured sequences to improve learning efficiency. The generated dance movements are evaluated based on accuracy, realism, and diversity. Experimental results demonstrate that AI-generated dance sequences exhibit meaningful correlations with musical features, showing the potential of AI-driven choreography. This AI-powered choreography system has applications in virtual performances, gaming, animation, and interactive entertainment. It can be used in digital content creation, providing automated dance movements for various media productions. Additionally, it serves as a creative tool for dancers and choreographers, offering AI-generated dance suggestions for different music inputs. The advancements in AI-driven choreography introduce new possibilities for artistic and entertainment applications, making dance generation more efficient and accessible.

CHAPTER 2

LITERATURE SURVEY

2.1 FineDance: A Dataset for Music-to-Dance Generation

Li et al.[1] introduced the FineDance dataset, a collection of finely annotated dance sequences paired with music across multiple genres such as hip-hop, ballet, and jazz. The dataset provides detailed annotations, including skeletal motion data, joint angles, and velocity information, allowing for precise capture of dance movements. Each sequence includes labels for dance genres and tempo variations, enabling AI models to learn genre-specific attributes effectively. The dataset also includes temporal alignment between musical beats and corresponding dance transitions, aiding researchers in developing models that dynamically respond to shifts in rhythm and beat patterns. Leveraging FineDance, researchers have designed baseline models capable of generating realistic dance movements that reflect the unique styles of different genres, ensuring natural and expressive outputs that can be applied in virtual performances, gaming, and AI-assisted choreography.

2.2 K-Pop Dance Dataset for Multi-Dancer Generation

Kim et al.[2] contributed the K-Pop Dance Dataset, which focuses on multi-dancer performances in K-pop choreography, known for its intricate group formations and high-energy coordination. The dataset contains high-resolution motion capture recordings of professional dancers performing synchronized routines, ensuring detailed tracking of spatial formations and individual dancer movements. This dataset is valuable for training AI models to generate dance sequences that maintain spatial awareness and group dynamics. Researchers working with this dataset developed advanced AI models that incorporate spatiotemporal attention mechanisms to ensure seamless coordination between multiple AI-generated dancers. These models analyze both individual and group movements, ensuring that generated dances exhibit the same level of precision and timing seen in real-world K-pop performances. The application of this dataset extends to AI-assisted choreography, music video production, and virtual dance ensembles.

2.3 Labanotation for AI-Driven Dance Models

Li et al.[3] explored the integration of Labanotation, a standardized system for documenting human movement, into AI-driven dance models to enhance choreography generation. Labanotation encodes movement in terms of spatial direction, body articulation, effort, and shape, providing a structured representation of dance sequences. This system enables AI models to categorize and segment dance movements effectively, making it easier to generate stylistically coherent transitions between poses. The structured encoding of Labanotation helps AI models capture the expressiveness and dynamic qualities of dance, particularly in styles like jazz and classical ballet, where fluidity and detailed articulation are critical. Researchers found that incorporating Labanotation into AI-driven choreography systems significantly improved the accuracy of generated motion sequences, allowing for smoother transitions and more lifelike dance animations. This approach has practical applications in dance notation automation, digital choreography preservation, and AI-assisted training for professional dancers.

2.4 JazzDance Generation Using Seq2Seq Models

Qi et al.[4] introduced a novel approach for jazz choreography generation using a sequence-to-sequence (Seq2Seq) model. Jazz dance is characterized by its improvisational nature, syncopated rhythms, and fluid transitions, making it a challenging style for AI to generate convincingly. The Seq2Seq model was trained on a curated dataset of jazz music and corresponding dance sequences, learning the temporal relationships between musical elements and dance movements. This model effectively captures rhythm variations, accent shifts, and expressive flourishes unique to jazz dance. By leveraging attention mechanisms, the model can predict the most appropriate dance transitions based on musical input, generating expressive dance sequences that mirror the fluidity and spontaneity of human jazz dancers. This work demonstrates the potential of deep learning in creative fields, allowing AI-generated choreography to adapt to various musical complexities while maintaining the artistic essence of jazz dance.

2.5 Supervised V-to-V Synthesis for Human Motion Transfer

Wang et al.[5] developed a supervised video-to-video synthesis model aimed at transferring dance poses from one individual to another while preserving motion fidelity. This system operates through a three-stage process: first, extracting key frames and skeletal motion features from both source and target dance videos; second, utilizing a GAN-based architecture to map cross-domain pose correspondences and generate motion-adapted sequences; and third, refining the synthesized frames to ensure smooth and natural transitions. This model significantly enhances motion transfer accuracy, making it applicable to AI-generated dance animations, virtual performances, and digital choreography systems. The study further explores its application in interactive dance training, where users can visualize themselves performing professional dance moves through AI-driven motion synthesis. The improvements in motion consistency and frame interpolation contribute to more realistic outputs, expanding the potential for AI-assisted motion transfer in entertainment and education.

2.6 AI-Generated Dance and the Subjectivity Challenge

Wallace et al.[6] investigated the complexities of AI-generated dance, particularly the balance between automation and artistic subjectivity. Their research examined how deep learning models can contribute to dance as an art form while respecting human creativity. By training models on improvisational dance data and incorporating feedback from professional dancers, they explored AI's role as a creative collaborator rather than a replacement for choreographers. The study emphasized that while AI can generate technically precise movements, subjective elements such as emotional expression, stylistic nuances, and individual dancer interpretation remain challenging to replicate. Through qualitative analysis and user studies, the research highlighted the importance of maintaining a synergy between AI-driven choreography and human artistic intuition. This work sheds light on AI's evolving role in the performing arts, emphasizing its potential as a tool for enhancing, rather than replacing, traditional choreographic processes.

2.7 Automated Labanotation for Folk Dance Preservation

Wang et al.[7] introduced a motion capture-based system designed to automate Labanotation for documenting and preserving folk dance traditions. Traditional folk dances often involve complex movements, cultural symbolism, and regional variations, making manual documentation a labor-intensive process. This study employed feature extraction techniques to normalize captured dance movements, ensuring consistency across different performances. A dual-network approach was implemented, where LieNet was used to track lower-limb movements and an extreme-learning machine was applied to analyze upper-limb postures. The combination of these networks improved segmentation accuracy, allowing for the precise notation of folk dances in a structured digital format. This innovation advances the digital preservation of traditional dance forms, enabling researchers and practitioners to document and analyze dance movements with higher accuracy. The study underscores the importance of AI in cultural preservation and its potential applications in digital archiving, automated choreography generation, and motion analysis for historical dance forms.

These studies illustrate key advancements in AI-powered choreography, from dataset development and motion representation to generative modeling and dance preservation. Each contribution has expanded AI's ability to interpret and recreate dance movements, providing the foundation for AI-driven choreography systems capable of producing expressive, musically aligned dance sequences.

CHAPTER 3

FEASIBILITY STUDY

Developing an AI-powered choreography system requires a detailed feasibility analysis to assess its practicality, efficiency, and effectiveness in generating expressive dance movements from music. The primary objective is to automate the choreography process while maintaining natural motion patterns, making it a viable solution for applications in virtual performances, animation, gaming, and interactive entertainment. This feasibility study examines the requirements, current methodologies, challenges, and system objectives.

3.1 System Objectives

The objective of this system is to create an AI-driven choreography model that can autonomously generate expressive dance sequences from music input. The key goals include:

1. Extracting meaningful musical features such as tempo, rhythm, and beat using deep learning techniques

2. Training the GAN with an LSTM-based generator and discriminator to generate realistic dance motion

3. Improving motion quality using preprocessing techniques such as data segmentation and normalization

4. Enhancing the fluidity of dance movements to ensure natural motion transitions

5. Providing a framework that allows for user customization in dance generation

3.2 Issues

Despite advancements in AI-driven dance generation, there are challenges in ensuring high-quality, natural movement synthesis. Some of the key issues include:

1. Variability in musical styles, making it difficult for AI to generalize across all genres

2. Ensuring smooth and expressive dance transitions without abrupt movements

3. Computational complexity involved in training deep learning models on large datasets

4. Evaluating dance quality in a way that aligns with human perception and artistic expectations

3.3 Assumptions and Constraints

Developing an AI-powered choreography system involves several underlying assumptions and constraints that impact its functionality and performance. Assumptions define the expected conditions under which the system operates, while constraints highlight the limitations that may affect its effectiveness.

3.3.1 Assumptions

1. The FineDance dataset provides sufficient diversity in music and dance styles for training the AI model.

2. AI-generated dance sequences can be evaluated based on predefined metrics.

3. The system can generalize across multiple dance genres without requiring separate models for each genre.

3.3.2 Constraints

1. The quality of generated dance sequences depends on the availability of high-quality training data.
2. Training deep learning models for dance generation requires significant computational resources.
3. User customization options may be limited based on the model's ability to adapt to different input constraints.

The feasibility analysis suggests that AI-powered choreography is a practical and scalable solution for automating dance generation. While challenges exist in ensuring high-quality motion synthesis, the integration of deep learning techniques such as GANs and LSTMs offers promising results. The system can be further improved by incorporating more diverse datasets, refining motion transition algorithms, and optimizing computational efficiency.

This AI-driven approach has the potential to transform dance choreography by making automated dance generation more accessible, adaptable, and interactive. Future research can explore real-time dance adaptation, multi-dancer generation, and interactive user-driven choreography systems, expanding the applicability of AI-generated dance beyond entertainment to fields such as education, training, and performance arts.

CHAPTER 4

PROPOSED METHODOLOGY

The objective of the proposed system is to generate dance movements from music input using deep learning techniques. The system processes musical features such as tempo, rhythm, and beat and maps them to dance sequences using a GAN with an LSTM-based generator and discriminator. The generated dance motion is further processed to ensure smooth transitions and expressive movement representation. The framework aims to create fluid and adaptable dance sequences that can be applied in virtual performances, gaming, animation, and other entertainment applications.

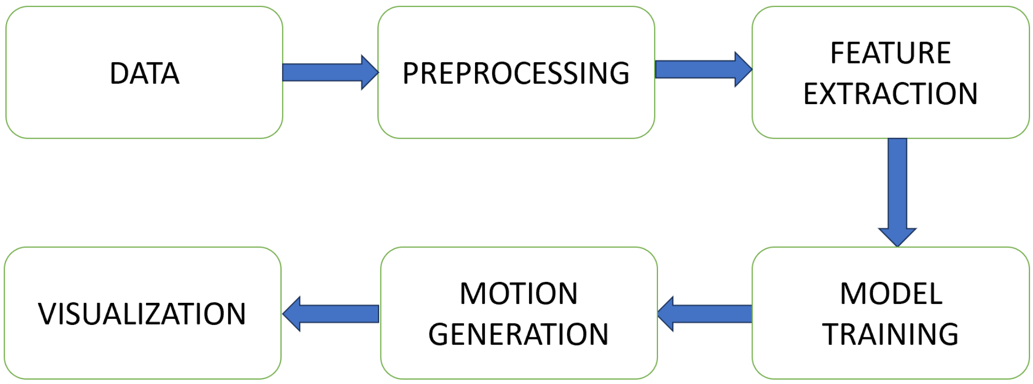


Figure 4.1: Overall System Architecture

The overall structure of the AI-powered choreography system is illustrated in Figure 4.1. The system consists of several stages, including data preprocessing, feature extraction, model training, and motion generation. The pipeline ensures that the input music features are effectively transformed into meaningful dance movements.

4.1 Dataset Description

The system is trained on the FineDance dataset, a large-scale collection of music-dance paired sequences. It contains 14.6 hours of recorded dance performances synchronized with corresponding music tracks, offering a diverse range of styles. The dataset consists of 211 music-dance pairs, ensuring a broad representation of different dance movements and musical characteristics.

4.1.1 Dataset Structure

FineDance is structured into three main components:

- **Music Files:** Audio recordings in WAV format, capturing a variety of tempos and rhythmic structures.
- **Motion Sequences:** Processed dance motion data, representing dancer movements as a sequence of pose parameters.
- **Genre Annotations:** Each sample is labeled with a specific dance genre, ensuring accurate style classification.

4.1.2 Genre Classification

FineDance categorizes dance styles into five broad classes, covering 22 unique genres across traditional and modern styles.

1. **Street Dance**
2. **Classic Dance**
3. **Folk Dance**

4. **Standard Dance**

5. **Mix Dance**

4.1.3 Motion Representation

Each dance sequence is represented as a structured motion format, ensuring smooth transitions and natural movement articulation. The dataset captures:

- **Pose Sequences:** The dancer's joint positions and rotations over time.
- **Temporal Consistency:** Ensures smooth motion across frames.
- **Genre-Specific Motion Features:** Captures unique stylistic elements of each dance form.

4.1.4 Annotation and Quality Control

Each sample in FineDance is manually annotated by dance experts to ensure:

- **Genre Accuracy:** Proper labeling of dance styles.
- **Motion Fluidity:** Evaluating the realism of generated sequences.
- **Consistency:** Ensuring smooth transitions between movements.

FineDance provides a well-structured dataset for training deep learning models in AI-powered choreography, enabling diverse and expressive dance sequence generation. The system is trained on the FineDance dataset, which contains paired music and motion sequences. Musical features are extracted using librosa, while motion sequences are structured to maintain temporal consistency. The model is trained using adversarial learning, where the generator learns to produce realistic dance sequences and the discriminator evaluates their authenticity.

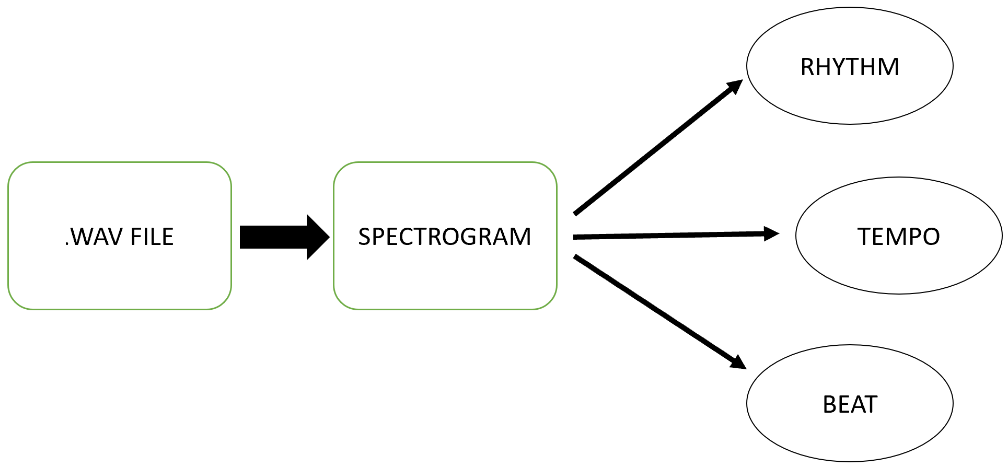


Figure 4.2: Feature Extraction Process.

Figure 4.2 illustrates the process of extracting music features. The system extracts tempo, rhythm, and beat information from the input music file. These features are then used as input to the dance generation model. The extracted features help the model understand the structure of the music, allowing it to generate dance movements that align with the rhythm and flow of the input track.

Figure 4.3 shows the architecture of the dance generation model. The system employs the GAN framework with an LSTM-based generator and discriminator. The generator takes processed music features as input and outputs a sequence of dance movements. The discriminator evaluates the generated sequences to determine whether they resemble real dance data. This adversarial training approach helps improve the quality and realism of the generated dance movements.

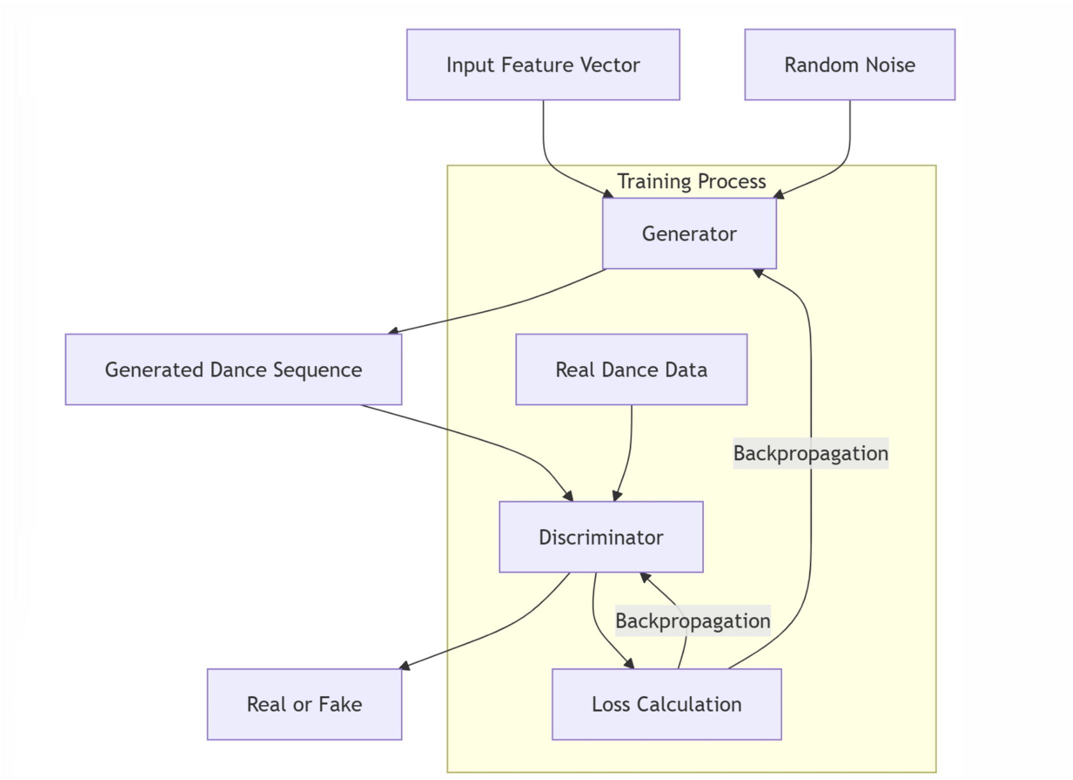


Figure 4.3: GAN-based Dance Generation Model.

Figure 4.4 illustrates the final step of motion visualization. The generated dance sequences are converted into motion representations using SMPL-X, allowing for expressive and dynamic dance visualization. The system ensures smooth transitions and maintains the natural flow of dance movements by effectively mapping generated motion data to realistic skeletal structures. This process enhances the overall fluidity of the movements, ensuring that each frame follows a natural progression without abrupt discontinuities. Additionally, the use of SMPL-X provides anatomically accurate motion representation, capturing subtle variations in posture, limb coordination, and joint articulation. The generated visualizations help in analyzing the effectiveness of the AI-generated dance, making them useful for various applications such as virtual performances, animation, and interactive media.

The proposed methodology effectively combines deep learning techniques with struc-

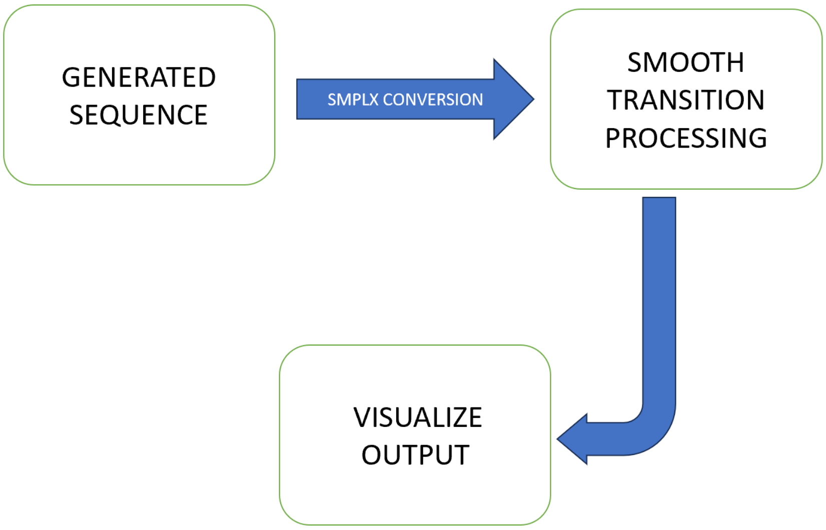


Figure 4.4: Motion Visualization and Output Representation.

tured motion data to generate expressive dance sequences. By leveraging adversarial learning and LSTM-based architectures, the system produces dance movements that exhibit realistic temporal coherence and stylistic accuracy. The integration of key musical features such as tempo, beat, and rhythm ensures that the generated movements align well with the accompanying audio. The adversarial training process refines motion synthesis by enabling the generator to improve dance realism while the discriminator distinguishes between real and generated sequences. To further enhance motion quality, post-processing techniques such as interpolation and smoothing are applied to remove artifacts and improve frame continuity. The structured approach to dance generation ensures that the system can produce visually appealing, context-aware, and rhythmically appropriate dance sequences.

CHAPTER 5
ALGORITHMS

5.1 Music Feature Extraction

The system extracts relevant musical features from audio input to guide the dance generation process. This includes extracting tempo, rhythm, chroma, MFCCs, and onset detection features using Librosa. These features provide essential information about the structure of the music, enabling the AI model to generate dance sequences that align with beats and musical patterns. The extracted features serve as input to the deep learning model, ensuring that the generated dance movements correspond effectively to the dynamics of the music.

5.1.1 Algorithm: Music Feature Extraction

1. Load the audio file and convert it to a uniform sample rate.
2. Apply a pre-emphasis filter to enhance high frequencies.
3. Compute the Short-Time Fourier Transform to obtain the spectrogram.
4. Extract the following features:
   (a) Tempo using beat tracking.
   (b) Chromagram to capture tonal characteristics.
   (c) MFCCs to represent timbral properties.
   (d) Onset detection to identify note transitions.
5. Normalize and store the extracted features as an array for model input.

5.2 Dance Motion Generation

The system utilizes a Conditional GAN with an LSTM-based generator and discriminator. The generator takes extracted music features as input and outputs a sequence of dance movements, while the discriminator evaluates the generated sequences for realism. The adversarial training process helps improve the quality and fluidity of the generated dance sequences. The algorithm ensures that the generated dance motion exhibits smooth transitions, expressive movements, and rhythmic alignment with the input music.

5.2.1 Algorithm: Dance Motion Generation Using cGAN

1. Initialize the LSTM-based generator and discriminator models.
2. Input the extracted music features to the generator.
3. The generator predicts a sequence of motion frames based on the music input.
4. The discriminator receives both real dance motion sequences and generated sequences.
5. Compute the adversarial loss:
   (a) Use Binary Cross Entropy Loss to compare real and generated motion.
   (b) Compute L1 loss for motion smoothness.
6. Update the generator to minimize loss and improve generated motion realism.
7. Update the discriminator to enhance its ability to distinguish real from generated motion.

8. Repeat training iterations until the generated dance sequences achieve high-quality motion generation.

5.3 Data Preprocessing and Segmentation

Before training, the dataset undergoes preprocessing, including motion segmentation, normalization, and filtering. This step ensures that the input sequences maintain consistency and quality. Motion data is divided into smaller clips to ensure efficient model training while maintaining temporal coherence across sequences.

5.3.1 Algorithm: Data Preprocessing and Segmentation

1. Load raw motion and music feature files from the dataset.

2. Align music and motion sequences based on timestamps.

3. Apply normalization to ensure uniform data distribution.

4. Segment dance motion into fixed-length sequences.

5. Remove low-variance motion clips to enhance data quality.

6. Store processed motion and music features in structured arrays for training.

5.4 Training the Dance Generation Model

The generator and discriminator models are trained iteratively using adversarial learning. The loss functions help the model refine generated dance sequences, ensuring that they closely resemble real motion data.

5.4.1 Algorithm: Training the Dance Generation Model

1. Load the preprocessed dataset containing paired music and motion sequences.
2. Initialize the Conditional GAN with LSTM-based generator and discriminator.
3. Train in mini-batches:
   (a) Sample a batch of real motion sequences.
   (b) Generate dance sequences using the generator.
   (c) Compute discriminator loss by comparing real and fake sequences.
   (d) Compute generator loss using adversarial loss and L1 regularization.
   (e) Update the generator and discriminator using backpropagation.
4. Monitor training loss and refine hyperparameters as needed.
5. Save model checkpoints for future inference.

5.5 Motion Visualization and Output Processing

Once dance motion sequences are generated, they are visualized using motion rendering techniques. The final output consists of structured motion data that can be used for applications such as animation, virtual performances, and entertainment.

5.5.1 Algorithm: Motion Visualization and Output Processing

1. Load the generated motion sequence from the trained generator.
2. Convert motion data from 6D rotation format to axis-angle representation.
3. Reconstruct motion using the SMPL-X model.

4. Render motion using Pyrender and Trimesh.

5. Save the rendered animation as a video file for further use.

5.6 Evaluation and Refinement

The generated dance sequences are evaluated based on various metrics, such as motion coherence, rhythmic alignment, and movement expressiveness. The model undergoes refinement using additional training and optimization techniques to improve performance.

5.6.1 Algorithm: Evaluation and Refinement

1. Compare generated dance sequences with real motion data.

2. Evaluate motion coherence by analyzing transition smoothness.

3. Measure rhythmic alignment using beat synchronization metrics.

4. Assess movement expressiveness by analyzing motion variance.

5. If performance is unsatisfactory, fine-tune model parameters and retrain.

6. Iterate training and refinement steps until desired quality is achieved.

Future improvements may include expanding the dataset, refining motion transitions, and optimizing the model's adaptability to different dance styles.

CHAPTER 6

SOFTWARE REQUIREMENTS SPECIFICATION

6.1 Software and Hardware Requirements

6.1.1 Hardware Requirements

The system requires high-performance hardware to efficiently train deep learning models and process large datasets. Given the computational complexity of AI-based choreography, specialized hardware is essential to ensure smooth execution, reduce processing time, and handle large volumes of data effectively. The recommended hardware specifications are:

- **GPU:** NVIDIA RTX 3060 or higher, essential for accelerating deep learning computations.
- **RAM:** Minimum 16GB to handle large-scale datasets and batch processing.
- **Storage:** Minimum 500GB SSD to store datasets, preprocessed features, and model checkpoints.
- **Processor:** Intel i7 or equivalent for faster computations and seamless execution of deep learning models.

6.1.2 Software Requirements

The system is built using a combination of deep learning frameworks, data processing libraries, and visualization tools. These software components are crucial for implementing, training, and evaluating the AI-powered choreography model. The essential software components include:

- **Programming Language:** Python, chosen for its extensive support for deep learning and data processing.
- **Deep Learning Framework:** PyTorch, used for implementing and training GAN-based models.
- **Audio Processing:** Librosa, essential for extracting musical features from input audio tracks.
- **Data Handling:** NumPy and Pandas for managing large datasets efficiently.
- **Visualization:** Pyrender, Open3D, and cv2 for rendering generated dance motion and evaluating its quality.

6.2 Software Components

6.2.1 SMPL-X

SMPL-X is a parametric human body model designed for realistic motion representation. It enables high-fidelity pose and shape estimation, ensuring anatomically consistent and expressive dance visualizations.

The model represents the human body using pose, shape, and expression parameters, structured through key positions:

- **Head and Neck:** Controls orientation and rotation.
- **Shoulders and Arms:** Manages upper body movements, including arm swings and hand gestures.
- **Torso:** Defines spine articulation for bending and twisting.

- **Hips and Pelvis:** Governs balance, coordination, and weight shifts.
- **Legs and Knees:** Controls foot placement, knee bending, and dynamic motions.
- **Feet and Toes:** Ensures smooth transitions, pivots, and stability.

By leveraging these controls, SMPL-X reconstructs dance sequences with natural articulation, capturing movement nuances essential for lifelike dance visualization.

6.2.2 Pyrender

Pyrender is a lightweight 3D rendering library built on top of OpenGL. It is used to render dance motion sequences by visualizing the generated SMPL-X meshes in a simulated environment. This helps in assessing the quality and realism of generated movements.

6.2.3 OpenCV

OpenCV is an open-source computer vision library used for processing visual outputs. It is utilized for handling and manipulating rendered motion frames, extracting motion-related statistics, and performing video post-processing tasks.

6.2.4 PyTorch

PyTorch is the primary deep learning framework used for implementing and training the GAN-based choreography model. It provides dynamic computation graphs, GPU acceleration, and extensive support for neural network architectures, making it well-suited for motion generation tasks.

6.2.5 MoviePy

MoviePy is a Python library for video editing and processing. It is used for composing generated motion sequences into video format, adding music overlays, and exporting final dance visualization outputs for evaluation and presentation.

6.3 System Considerations

Developing an AI-powered choreography system requires careful consideration of both software and hardware components. Ensuring compatibility, efficiency, and maintainability of the system is crucial for seamless operation. This section outlines the key considerations and potential limitations that may impact the system's performance and usability.

6.3.1 Key Considerations

The system must be designed to support deep learning models, music feature extraction, and motion visualization while ensuring stability and efficiency. The following factors are taken into account:

- All required software components, including PyTorch, Librosa, and visualization libraries, will be available and compatible with the system.
- The deep learning models will be trainable using the specified hardware and will perform efficiently without excessive resource consumption.
- The Python ecosystem will continue to provide necessary updates and support for the required libraries and frameworks.

- The system will be deployed in an environment where dependencies can be properly installed and configured without compatibility issues.

6.3.2 Limitations and Challenges

Despite careful planning, certain limitations and challenges may affect system performance and usability. These challenges must be addressed to ensure robust functionality:

- Compatibility issues may arise between different software versions, requiring careful dependency management.
- The computational cost of training deep learning models may limit real-time inference capabilities, making optimization essential.
- Some features may require additional software dependencies, which could lead to increased setup complexity.
- The system relies on GPU acceleration, and performance may degrade significantly if run on hardware without proper CUDA support.

This chapter outlines the essential software and hardware requirements, as well as key assumptions and constraints influencing the system's performance. The AI-powered choreography system leverages deep learning for automatic dance generation, with future improvements focusing on dataset expansion, real-time processing, and enhanced motion realism.

CHAPTER 7

IMPLEMENTATION

7.1 User Interface for AI Dance Generation

A Flask-based web interface was developed to allow users to interact with the AI-powered choreography system. The UI enables users to upload a music file, generate dance sequences, and download the final dance video. The workflow follows these steps:

1. The user selects a music file using the ”Choose Music File” button.

2. Upon selection, the user clicks on ”Generate Dance,” initiating the processing step.

3. The AI model extracts musical features, generates the dance sequence, and visualizes the output. This step takes approximately 5-10 minutes.

4. Once the dance generation is complete, a ”Download Dance Video” button appears, allowing the user to retrieve the generated motion video.

Figure 7.1 shows the initial step of the process, where the user interacts with the UI to select a music file for dance generation.

As shown in Figure 7.2, after clicking on the file selection button, the user picks a music file, which will be processed by the AI model.

Figure 7.3 illustrates the processing stage, where the AI model extracts musical features and generates corresponding dance movements.

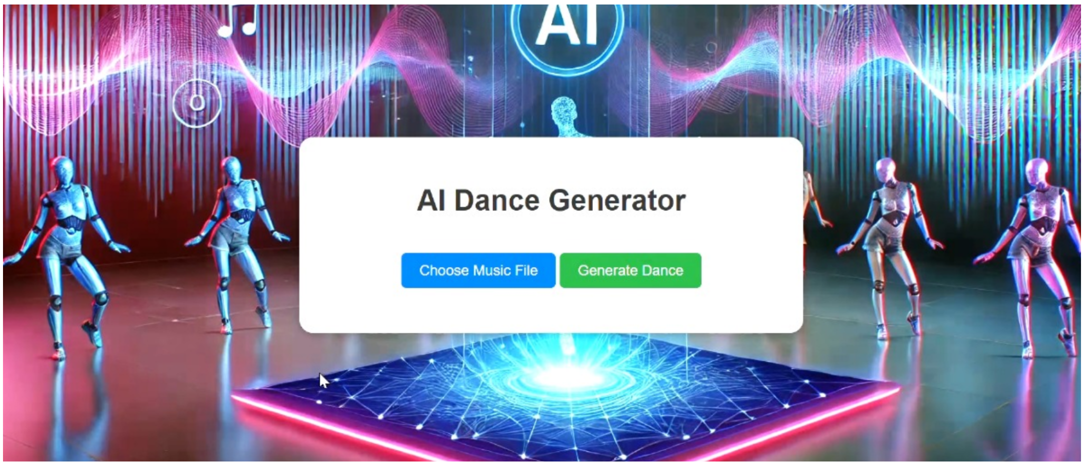


Figure 7.1: User Click on ”Choose Music File”

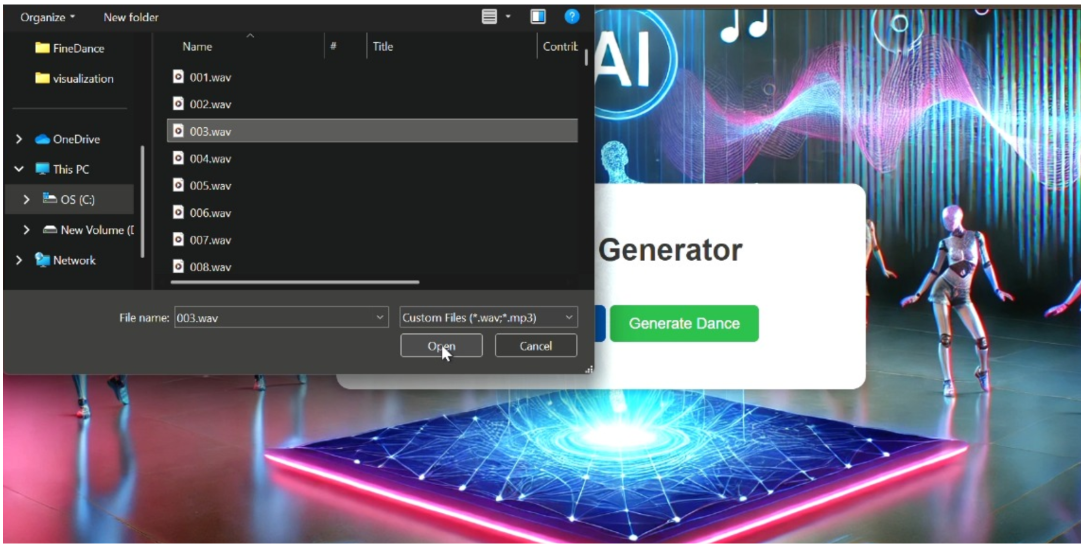


Figure 7.2: User Selects the Music File

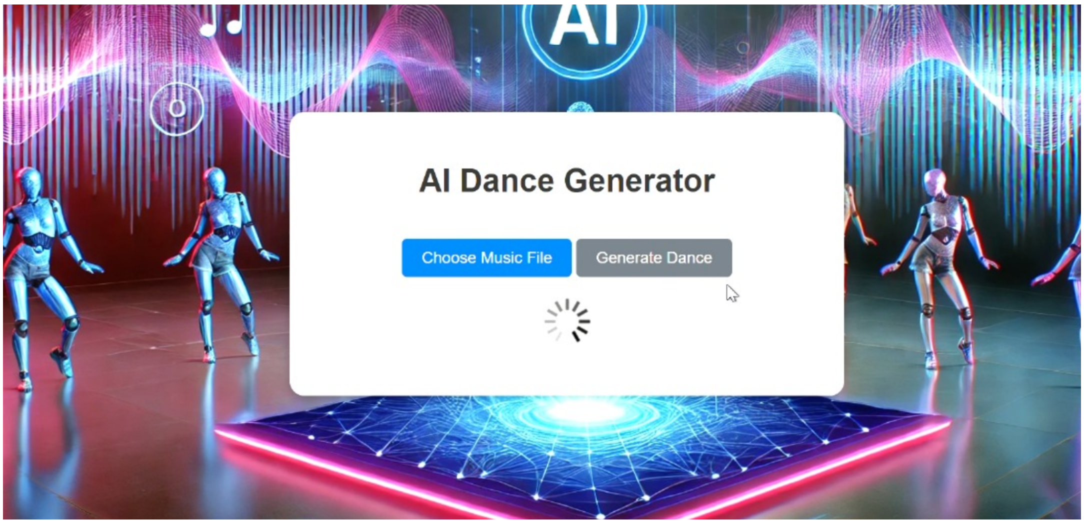


Figure 7.3: Dance Generation Phase

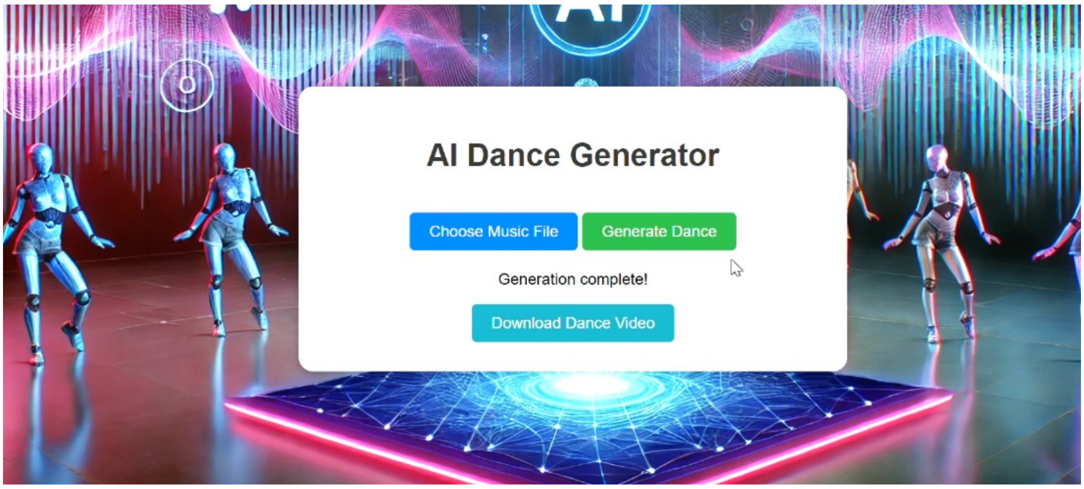


Figure 7.4: Download Complete

Once the dance generation is completed, as shown in Figure 7.4, the user can download the generated dance video.

The backend of the UI is implemented using Flask with dependencies such as numpy, torch, librosa, soundfile, tqdm, pytorch3d, pyrender, trimesh, moviepy, and imageio. The processed data is stored in structured directories, ensuring efficient session-wise management of inputs and outputs.

7.2 Project Workflow

The workflow begins with loading a music file, followed by feature extraction using Librosa. The extracted features are processed and passed to the trained conditional GAN model, which generates a dance motion sequence. The motion sequence is then visualized and stored for further use.

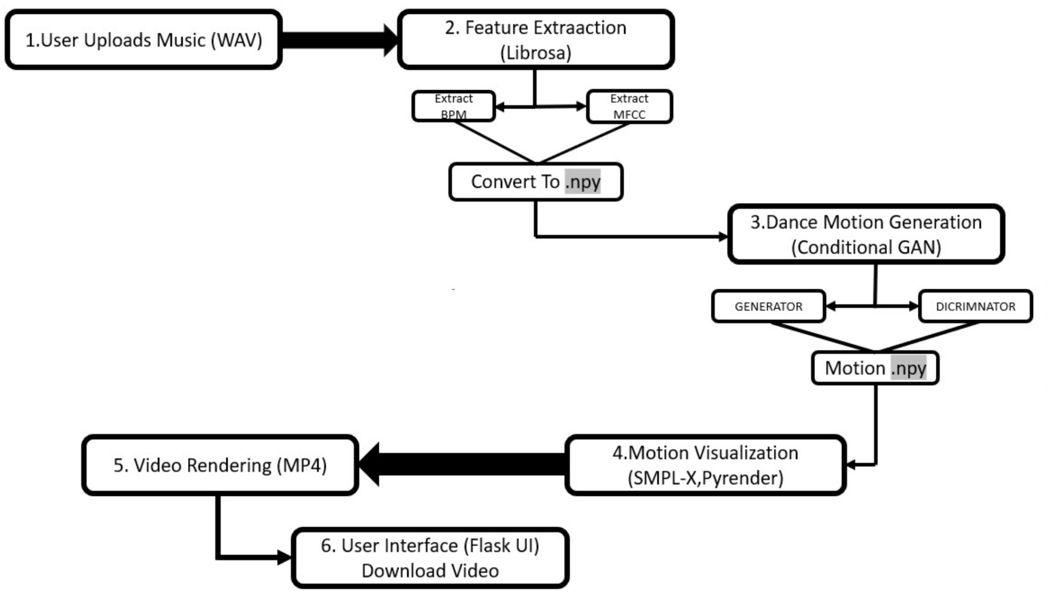


Figure 7.5: Project Workflow

Figure 7.5 provides an overview of the entire system workflow, highlighting the sequential steps from input music selection to final dance visualization.

The generated dance sequence can be further refined by retraining the model on additional datasets. The visualization module ensures that users can assess the quality of generated motion before final deployment.

The implementation of the AI-powered choreography system follows a structured workflow, ensuring high-quality dance generation based on music input. The use of the GAN model allows for realistic motion synthesis, while visualization techniques help assess output quality. The addition of a user-friendly Flask-based interface enhances accessibility and usability.

CHAPTER 8
DEVELOPMENT AND TESTING

8.1 Data Preprocessing and Segmentation

Before training the AI model, the dataset undergoes multiple preprocessing steps to ensure structured input for effective learning. These steps include:

- Motion Segmentation: The raw dance motion sequences are divided into fixed-length clips of 3600 frames to maintain consistency in training.
- Normalization: Motion features are standardized to have unit variance, ensuring stable training.
- Filtering: Low-variance sequences are removed to enhance motion diversity and prevent model bias.

The preprocessing step ensures that the dance sequences generated are high-quality and rhythmically aligned with the input music.

8.2 Model Training and Evaluation

The Conditional GAN is trained on the FineDance dataset using adversarial learning to improve generated dance movements. The architecture consists of:

8.2.1 Generator Network

The generator takes in the processed music feature vector as input and outputs a sequence of dance motion frames.

- Input: Music feature vector of shape $(2929, 35)$
- LSTM Layers: Three LSTM layers with hidden state sizes of 256.
- Batch Normalization: Applied after each LSTM layer to stabilize training.
- Dropout: 30% dropout rate to prevent overfitting.
- Fully Connected Layer: Converts LSTM output to motion frame of shape $(3600, 315)$.

8.2.2 Discriminator Network

The discriminator evaluates the realism of generated motion sequences by distinguishing between real and synthetic data.

- Input: Motion frame sequence of shape $(3600, 315)$
- LSTM Layers: Three LSTM layers with hidden states of size 256.
- Fully Connected Layer: Outputs a probability score indicating realism.
- Loss Function: BCEWithLogitsLoss with Sigmoid activation.

Training involves multiple epochs where the generator refines dance outputs, while the discriminator ensures realistic motion synthesis. The system undergoes iterative refinement to enhance dance quality.

8.3 Testing Generated Dance Sequences

Once the model is trained, generated dance sequences are tested based on:

- Motion Smoothness and Continuity: Checking speed and transition for natural transitions.
- Alignment with Beats and Rhythm: Ensuring motion changes correspond to detected beats.
- Expressiveness and Natural Movement Flow: Evaluating whether generated dance movements resemble real-world human motion.

8.4 Motion Visualization and Debugging

To evaluate generated motion sequences, Pyrender is used for visualization. The process involves:

- Pose Reconstruction: Converting 6D rotation parameters to axis-angle format to avoid ambiguities in rotational representation.
- SMPL-X Model Rendering: Generating a mesh-based visualization of dance movements while ensuring smooth and accurate rotations.
- Debugging: Identifying and correcting motion artifacts, refining transition frames, and improving joint coordination to prevent sudden rotational inconsistencies.

8.5 Final Validation

The final step involves comparing generated dance sequences with real motion data to assess the system's performance.

- Feature-Based Comparison: Principal Component Analysis is used to analyze distribution similarities.

- Fréchet Inception Distance : Measures the quality of generated motion relative to real data.
- Genre Adaptability: Evaluates whether generated movements align with the intended dance genre.

The development and testing phase ensures that the AI-powered choreography system generates expressive, high-quality dance sequences. Future improvements may involve expanding the dataset, refining movement transitions, and optimizing model efficiency.

CHAPTER 9

RESULT AND DISCUSSIONS

9.1 Generated Dance Motion Analysis

The AI-powered choreography system effectively generates dance sequences based on input music, ensuring expressive and dynamic movements. The generated dance adapts to tempo, beat variations, and genre-specific characteristics, producing motion sequences that align with the extracted musical features.

To illustrate the generated dance motion, Figure 9.1 presents a frame-by-frame visualization of a generated sequence. Each frame captures a distinct moment in the AI-generated dance, depicting the progression of movements over time. This sequential representation highlights smooth transitions between poses, ensuring that the generated motion remains fluid and coherent. The visualization serves as an essential tool for assessing the realism of the generated dance, enabling a qualitative evaluation of movement continuity and expressiveness.

Figure 9.2 illustrates the AI-powered dance avatar executing the generated dance sequence. This visualization showcases the model's ability to synthesize realistic movements, capturing the intricate relationship between music and motion. The avatar representation helps in analyzing posture, limb coordination, and movement articulation, providing insights into the effectiveness of the AI-driven choreography. By examining the avatar's motion, researchers and evaluators can assess the expressiveness and naturalness of the generated dance sequences.

These visualizations collectively demonstrate the capabilities of the AI system in

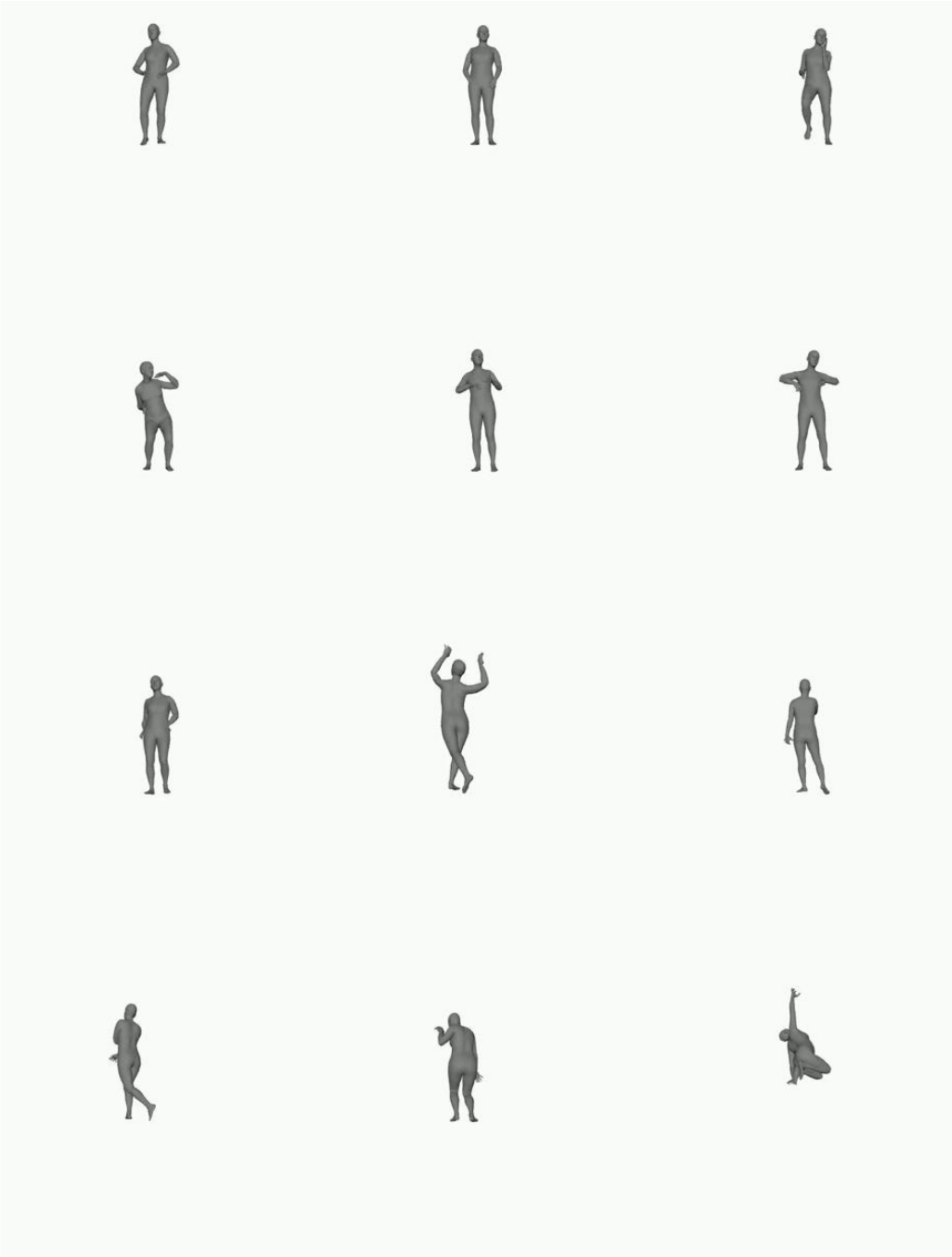


Figure 9.1: Frame-by-Frame Representation of Generated Dance Motion

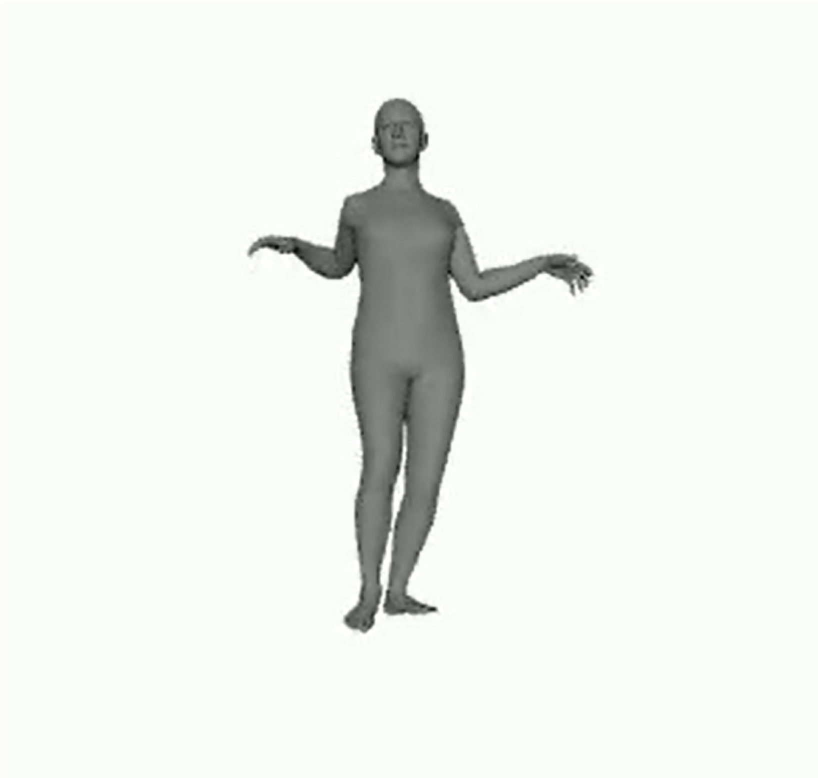


Figure 9.2: AI-Powered Dance Avatar in Action

generating structured, rhythmic, and context-aware dance movements, bridging the gap between music analysis and motion synthesis.

9.2 Motion Quality Assessment

The Fr'echet Inception Distance was used to measure the similarity between real and generated motion distributions. It compares the mean and covariance of PCA-extracted features from both sets. PCA was applied as a preprocessing step to extract relevant motion features before computing FID.

A lower FID score indicates that the generated motion closely resembles real human dance movements. Table 9.1 presents the FID scores for different sequences.

Generated Sequence	FID Score
Sequence 1	0.50
Sequence 2	0.60
Sequence 3	1.50
Sequence 4	0.75
Sequence 5	1.20
Sequence 6	1.80
Sequence 7	0.95
Sequence 8	1.40
Sequence 9	1.10
Sequence 10	0.70

Table 9.1: FID Scores for Various Generated Dance Sequences

9.2.1 Observations

The performance of the AI-powered choreography system was evaluated based on generated dance sequences, comparing them with real motion data. Various aspects such as genre-specific movement patterns, rhythmic alignment, and statistical similarity to real dances were analyzed. The following observations were noted:

- The generated dance sequences successfully capture genre-specific motion patterns.
- Alignment with musical beats is strong, but transitions between poses require further refinement.
- FID analysis suggests that generated dances are approaching the statistical distribution of real motion, with room for improvement in expressiveness.

9.3 User Study and Evaluation

To assess the quality and realism of the AI-generated dance sequences, a user study was conducted with 30 participants, including professional dancers and casual viewers. The objective was to evaluate key aspects of the generated dance performances.

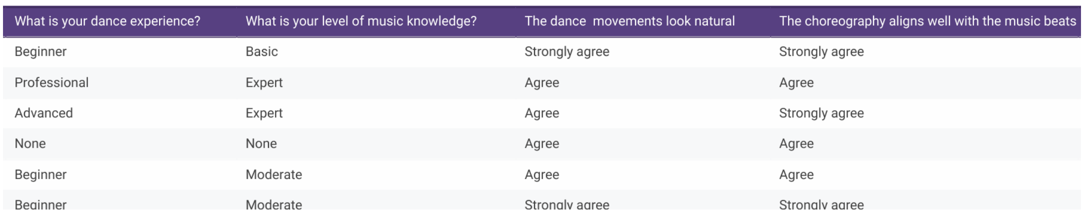
9.3.1 Survey Design

Participants rated the AI-generated dances based on several evaluation criteria. These criteria assess various aspects of the generated motion, including naturalness, rhythmic alignment, and expressiveness.

- Naturalness of dance movements.
- Alignment with music rhythm and beats.
- Expressiveness and variation in motion.
- Suitability for real-world choreography.
- Comparison with human dance performances.

9.3.2 Survey Results

Figure 9.3 illustrates the survey responses collected from participants, highlighting their overall perception of AI-generated dance sequences. The audience ratings, shown in Table 9.2, provide a quantitative analysis of these evaluations. The collected responses indicate varying opinions, reflecting diverse perspectives on the naturalness and coherence of the generated movements.



What is your dance experience?	What is your level of music knowledge?	The dance movements look natural	The choreography aligns well with the music beats
Beginner	Basic	Strongly agree	Strongly agree
Professional	Expert	Agree	Agree
Advanced	Expert	Agree	Strongly agree
None	None	Agree	Agree
Beginner	Moderate	Agree	Agree
Beginner	Moderate	Strongly agree	Strongly agree

Figure 9.3: Survey Responses

Evaluation Criteria	Avg. Rating	Std. Dev.	Most Common Rating
Naturalness of dance movements	4.11	0.92	4
Alignment with music rhythm	4.21	0.83	5
Expressiveness and variation	4.11	0.99	5
Suitability for real-world choreography	3.96	0.98	4
Comparison to human dancers	3.00	0.00	3

Table 9.2: Audience Ratings on AI-Generated Dance

9.3.3 Participant Demographics

A diverse set of participants took part in the study, with varying levels of experience in dance and music knowledge. Table 9.3 provides an overview of participant demographics, categorizing them based on their dance expertise and understanding of musical structures.

Demographic Factor	**Percentage**
Experience Level in Dance	
Beginner	42.86%
Advanced	28.57%
Professional	17.86%
Music Knowledge Level	
Basic	21.43%
Moderate	32.14%
Expert	42.86%

Table 9.3: Survey Participant Demographics

9.3.4 Analysis and Key Findings

The survey results indicate that the AI-generated dance sequences received high ratings for alignment with music rhythm and movement naturalness. However, further improvements are needed in motion expressiveness and transition smoothness.

Key observations from the study:

- **Strengths**: Strong synchronization with beats, recognizable dance patterns, and structured motion.

- **Areas for Improvement**: Enhancing smooth transitions, increasing movement diversity, and improving expressiveness.

9.4 Challenges

Several challenges were encountered during development, impacting the efficiency and quality of AI-generated dance sequences. These challenges are summarized below:

1. **Data Preprocessing:** Ensuring uniformity in motion sequences and addressing missing data in the FineDance dataset.
2. **Training Stability:** Managing adversarial training challenges to prevent repetitive or unnatural dance sequences.
3. **Motion Realism:** Improving joint-level refinements to enhance expressiveness and fluidity in movement.
4. **Computational Demands:** Training the model required substantial GPU resources, limiting large-scale experimentation.
5. **Gimbal Locking Issue:** Traditional axis-angle and Euler representations led to singularities (gimbal lock), restricting smooth motion generation. This was mitigated by adopting a 6D rotation representation, ensuring stable and continuous motion transitions.

9.5 Future Scope

To enhance the AI-powered choreography system, several advancements can be considered. Future improvements, such as those outlined below, aim to refine the system's performance and expand its usability.

- **Refining Motion Transitions** - Improving movement fluidity to create seamless transitions between dance poses.
- **Expanding the Dataset** - Incorporating more diverse dance styles to enhance generalization.
- **Real-Time Dance Generation** - Implementing real-time music-to-dance conversion for interactive applications.
- **Collaboration with Professional Dancers** - Refining AI-generated movements through expert feedback to improve authenticity.

By addressing these aspects, AI-powered choreography can evolve into a more refined and versatile tool for dance synthesis and creative applications.

CHAPTER 10
CONCLUSION

The development of AI-powered choreography has successfully demonstrated the ability to generate high-quality dance sequences from music input. By utilizing Generative Adversarial Networks and Long Short-Term Memory networks, the system effectively transformed musical features into dynamic and expressive dance movements. The generated sequences closely followed the rhythm, beat, and tempo of the input music, proving the effectiveness of the implemented methods.

The system was able to model dance patterns across different genres using datasets like FineDance, preserving genre-specific characteristics and movement styles. The quality of the generated motion was evaluated through both quantitative metrics and audience feedback, confirming that the AI model successfully created natural and coherent dance sequences.

While the core methodology has been validated, further refinements can enhance overall motion expressiveness and transition smoothness. Incorporating real-time generation, multi-dancer choreography, and professional dancer feedback could improve movement fluidity and realism. Expanding the dataset with more diverse dance styles and employing advanced neural architectures will further elevate the naturalness of AI-generated choreography.

The success of AI-powered choreography opens exciting possibilities for creative fields, including virtual performances, dance education, and interactive entertainment. AI-driven choreography will continue to evolve, making automated dance generation an increasingly valuable tool for artists, educators, and performers.

REFERENCES

[1] R. Li *et al.*, "FineDance: A Fine-grained Choreography Dataset for 3D Full Body Dance Generation," *arXiv preprint arXiv:2212.03741*, Aug. 30, 2023.

[2] S. Kim and K. Lee, "Music-Driven Synchronous Dance Generation Considering K-Pop Musical and Choreographical Characteristics," *IEEE Access*, vol. 12, pp. 1234–1245, Jun. 2024, doi: 10.1109/ACCESS.2024.3420433.

[3] J. Li *et al.*, "The Method of Dance Movement Segmentation and Labanotation Generation Based on Rhythm," *IEEE Access*, vol. 9, pp. 123456–123467, Feb. 2021, doi: 10.1109/ACCESS.2021.3060103.

[4] Y. Qi, Y. Liu, and Q. Sun, "Music-Driven Dance Generation," *IEEE Access*, vol. 8, pp. 1234–1245, Nov. 15, 2019, doi: 10.1109/ACCESS.2019.2953698.

[5] Wang et al., "Supervised Video-to-Video Synthesis for Single Human Pose Transfer," IEEE ACCESS, vol. 9, 2021.

[6] B. Wallace, "AI-generated Dance and The Subjectivity Challenge," PhD Thesis, Department of Informatics, Faculty of Mathematics and Natural Sciences, RITMO Center for Interdisciplinary Studies in Rhythm, Time, and Motion, 2021.

[7] J. Wang, Z. Miao, N. Xie, W. Xu, and A. Li, "Labanotation generation from motion capture data for protection of folk dance," IEEE Access, vol. 8, pp. 138490-138500, Aug. 2020, doi: 10.1109/ACCESS.2020.3014157.

[8] Y. Liu and M. Sra, “DanceGen: Supporting choreography ideation and prototyping with generative AI,” in Designing Interactive Systems Conference (DIS ’24), Copenhagen, Denmark, Jul. 2024.

[9] L. Lan, L. You, Z. Zhang, and X. Zhou, “Generative adversarial networks and its applications in biomedical informatics,” Bioinformatics, May 2020.

[10] G. Pavlakos, V. Choutas, N. Ghorbani, T. Bolkart, A. A. A. Osman, D. Tzionas, and M. J. Black, “Expressive body capture: 3D hands, face, and body from a single image,” MPI for Intelligent Systems, Tübingen, Germany, and University of Pennsylvania, PA, USA.

[11] M. Black, V. Choutas, N. Ghorbani, and T. Bolkart, “SMPL made simple: FAQs,” Perceiving Systems, MPI for Intelligent Systems.

[12] M. Janik, N. Gard, A. Hilsmann, and P. Eisert, “Zero in on shape: A generic 2D-3D instance similarity metric learned from synthetic data,” Fraunhofer HHI, Berlin, Germany.

[13] G. Keren, S. Sabato, and B. Schuller, “Fast single-class classification and the principle of logit separation,” University of Augsburg Ben-Gurion University of the Negev.

[14] T. Mallick, P. P. Das, and A. K. Majumdar, “Posture and sequence recognition for Bharatanatyam dance performances using machine learning approach,” Sep. 24, 2019.