Dissertation on

"Automated Dance Choreography and Direction Using Music Classification and Contextual Factors"

Submitted in partial fulfilment of the requirements for the award of degree of

Bachelor of Technology
in
Computer Science & Engineering

UE20CS390B – Capstone Project Phase - 2

Submitted by:

Rayudu Srishti	**PES1UG20CS329**
Vibha Murthy	**PES1UG20CS495**
Vidisha Chandra	**PES1UG20CS498**
Vishakha Hegde	**PES1UG20CS506**

Under the guidance of

Prof. Kathargуppe Srinivas
Professor
PES University

August - December 2023

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
FACULTY OF ENGINEERING
PES UNIVERSITY
(Established under Karnataka Act No. 16 of 2013)
100ft Ring Road, Bengaluru – 560 085, Karnataka, India

PES UNIVERSITY

(Established under Karnataka Act No. 16 of 2013)
100ft Ring Road, Bengaluru – 560 085, Karnataka, India

FACULTY OF ENGINEERING

CERTIFICATE

This is to certify that the dissertation entitled

'Automated Dance Choreography and Direction Using Music Classification and Contextual Factors'

is a bonafide work carried out by

Rayudu Srishti	**PES1UG20CS329**
Vibha Murthy	**PES1UG20CS495**
Vidisha Chandra	**PES1UG20CS498**
Vishakha Hegde	**PES1UG20CS506**

in partial fulfilment for the completion of seventh semester Capstone Project Phase - 2 (UE20CS390B) in the Program of Study - Bachelor of Technology in Computer Science and Engineering under rules and regulations of PES University, Bengaluru during the period August- December 2023. It is certified that all corrections / suggestions indicated for internal assessment have been incorporated in the report. The dissertation has been approved as it satisfies the 7th semester academic requirements in respect of project work.

Signature **Kathargуppe Srinivas** Professor	Signature Dr. Mamatha H R Chairperson	Signature Dr. B K Keshavan Dean of Faculty

External Viva

Name of the Examiners	**Signature with Date**
1. _____	_____
2. _____	_____

DECLARATION

We hereby declare that the Capstone Project Phase - 2 entitled **“Automated Dance Choreography and Direction Using Music Classification and Contextual Factors”** has been carried out by us under the guidance of Kathargуppe Srinivas, Professor and submitted in partial fulfilment of the course requirements for the award of degree of **Bachelor of Technology** in **Computer Science and Engineering** of **PES University, Bengaluru** during the academic semester August - December 2023. The matter embodied in this report has not been submitted to any other university or institution for the award of any degree.

PES1UG20CS329	**Rayudu Srishti**
PES1UG20CS495	**Vibha Murthy**
PES1UG20CS498	**Vidisha Chandra**
PES1UG20CS506	**Vishakha Hegde**

ACKNOWLEDGEMENT

We would like to express our gratitude to Prof. Katharguppe Srinivas, Department of Computer Science and Engineering, PES University, for his continuous guidance, assistance, and encouragement throughout the development of this UE20CS390B - Capstone Project Phase – 2.

We are grateful to the project coordinator, Dr Priyanka H, for organizing, managing, and helping with the entire process.

We take this opportunity to thank Dr. Mamatha H R, Chairperson, Department of Computer Science and Engineering, PES University, for all the knowledge and support we have received from the department. We would like to thank Dr. B.K. Keshavan, Dean of Faculty, PES University for his help.

We are deeply grateful to Dr. M. R. Doreswamy, Chancellor, PES University, Prof. Jawahar Doreswamy, Pro-Chancellor – PES University, Dr. Suryaprasad J, Vice-Chancellor, PES University for providing us with various opportunities and enlightenment every step of the way. Finally, this project could not have been completed without the continual support and encouragement we have received from our family and friends.

ABSTRACT

This project aims to simplify the choreography process for dancers and choreographers by creating an automated system that maps dance moves to the beats of a music piece. The system uses audio analysis, 3D animation, and contextual factors like background, setting, costume, and lighting to create a visually pleasing and immersive experience that accurately represents the musical characteristics of the input file.

The system begins by analyzing the tempo, rhythm, and tone of the music file to match it with the corresponding dance style. It generates a set of dance moves synchronized with the beats and rhythm of the music and takes into account the emotion of the song. The system then creates a 3D dancing figure that simulates the movements of a real-life dancer. The subsequent step involves crafting a 3D dancing figure that mirrors the fluidity and authenticity of real-life dancers. Additionally, the system goes beyond mere choreography, incorporating an analysis of contextual factors such as the overall mood of the music. It offers recommendations for enhancing the performance experience, encompassing aspects like lighting schemes, stage design, and costumes tailored to complement the visual and auditory elements. Through the automation of the choreography process, this system presents an efficient and user-friendly solution, empowering creators to produce visually and aurally immersive dance performances effortlessly.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER-1

INTRODUCTION

Dance is a universal form of expression that has been used throughout history to convey emotions, tell stories, and celebrate cultural traditions. With the advancement of technology, there has been an increasing interest in using Deep Learning (DL) techniques to generate dance movements automatically. One approach is to build a dance generation model that takes in a song as input, classifies it based on emotion, and maps the emotion to a seed motion to generate dance. In this project, we explore the feasibility of building such a model and evaluate its performance. Specifically, we collect a dataset of songs and corresponding dance motions that are annotated with emotional labels, preprocess the data, build a machine learning classifier to predict the emotions of the songs, map the predicted emotions to seed motions using a generative model, and generate dance sequences using a recurrent neural network. We also consider the dance style and generate an appropriate background setting for the dance, along with a suitable costume for the dancer based on the style and emotion of the song. The results of this project demonstrate the potential of automation techniques to generate dance movements that are expressive, creative, and engaging.

CHAPTER-2

PROBLEM STATEMENT

The challenge this project seeks to solve is the creation of captivating and expressive dance moves through the application of machine learning and artificial intelligence methods. Dance generation has advanced significantly; however, the majority of methods now in use generate motions according to pre-established choreography or rigid rules, which might restrict the variety and originality of the resulting dance sequences. Moreover, no models consider the dancing style or the emotional content of the music, which might lead to a mismatch between the movements and the performance's intended mood or genre. Furthermore, neither the suggested costumes nor the appropriate background for the dance are offered by these models to the user.

CHAPTER-3

LITERATURE REVIEW

This chapter provides an overview of the existing understanding in the field and examines significant discoveries that contribute to shaping, educating, and transforming our research.

3.1. [1] R. Li, S. Yang, D. A. Ross, A. Kanazawa, AI Choreographer: Music Conditioned 3D Dance Generation with AIST++ vis ICCV 2021

3.1.1. Summary:

"Dancing is a well-known artform known in all cultures, and generating realistic 3D dance movements from music is a challenging task". In this work, a "Full Attention Cross-modal Transformer" ("FACT") network is proposed to solve these challenges. The model generates a "long sequence of realistic 3D dance motions from a short seed motion and audio features". This "transformer" involves "three key design choices", including "full-attention mask," "N future motion prediction," and "cross-modal transformer." The "motion quality, generation diversity, and motion-music correlation" of the generated 3D dance motion are calculated using "Frechet Inception Distance" (FID), "average Euclidean distance," and "Beat Alignment Score," respectively.

3.1.2. Methodology:

The proposed “FACT” model “maps the seed motion and audio features” using two transformers namely, "motion transformer" and "audio transformer". These help convert the inputs into respective embeddings. The encoded embeddings are concatenated and sent to a "cross-modal transformer that generates motion sequences”. In each case it generates “N future motion sequences”. The model is trained in a “self-supervised manner” and applied in an auto-regressive framework at test time. In order to “FACT” uses "full-attention mask" and “predicts N future motions” beyond the current input to pay more attention to the “temporal context”.

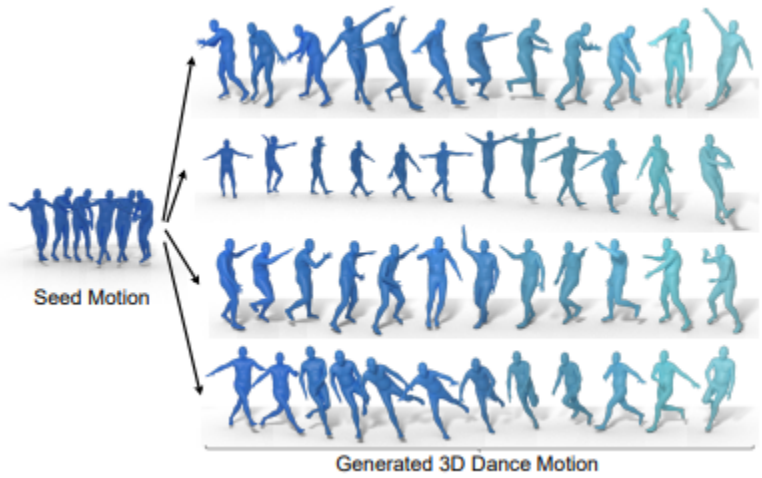

Figure 1

3.1.3. Evaluation:

The generated 3D dance motion is evaluated using FID, “average Euclidean distance”, and “Beat Alignment Score”. The “motion quality, generation diversity, and motion-music correlation” of the generated motion are compared with three baselines, including “GPT style causal transformer, motion encoder-decoder, and PCA-based method”. The evaluation results show that the proposed “FACT” model generates 3D dance motion with “high motion quality, generation diversity, and motion-music correlation”.

$$\text{BeatAlign} = \frac{1}{m}\sum_{i=1}^{m}\exp\left(-\frac{\min_{\forall t_j^y \in B^y}||t_i^x - t_j^y||^2}{2\sigma^2}\right) \quad (2)$$

Figure 2

3.1.4. **Shortcomings**:

The proposed “FACT” model does not reason about "physical interactions between the dancer and the floor”, “leading to artifacts such as foot sliding and floating”. Also, the model is currently "deterministic," and generating “multiple realistic dance motions per music is an exciting direction for future research.”

3.2. [2] Liu Xin, Ko Young Chun, The use of deep learning technology in dance movement generation vis 2022 Frontiers in Neurorobotics 16

3.2.1. Summary:

The model discussed is a basic mapping model that uses a generator to create “smooth dance postures” based on input music, and a “discriminator module to ensure consistency between the generated dance and the music”. The model also incorporates an “autoencoder module for the audio features”, and uses an “improved version of the Pix2PixHD model” to transform dance pose sequences into a “real-life dance”. The model has shown “superior performance to LSTM series models” and generates dances that are “closest to real postures”. However, the model may experience “video glitches due to network environment”, and the amount of data used is limited.

3.2.2. Methodology:

The model uses a “generator module to create smooth dance postures based on input music”, and a “discriminator module to ensure consistency between the generated dance and the music”. The audio features are processed using an autoencoder module.

The improved “Pix2PixHD model” is used to transform the “generated dance pose sequences into real-life dance sequences”.

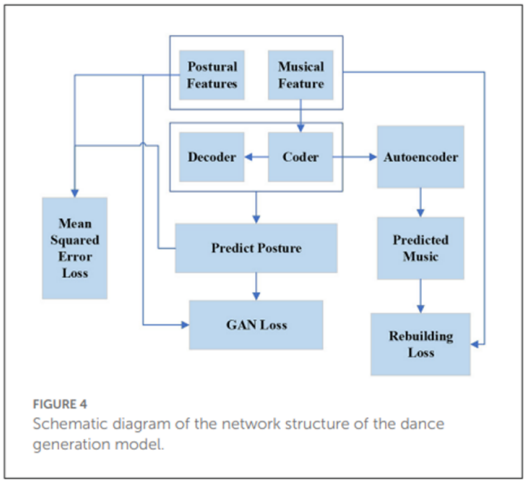

FIGURE 4
Schematic diagram of the network structure of the dance generation model.

Figure 3

3.2.3. Evaluation:

The model's performance is “evaluated based on the consistency between the generated dance and music” using a “discriminator module, and the quality of the generated dance using the improved Pix2PixHD model”. The model has shown “superior performance compared to LSTM series models” and generates dances that are “closest to real postures”.

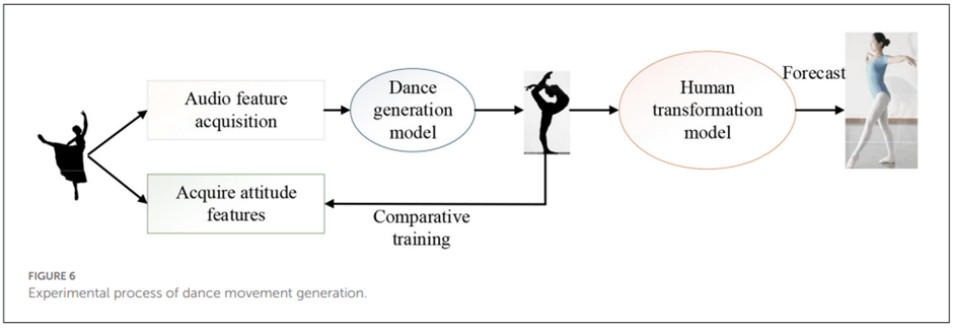

FIGURE 6
Experimental process of dance movement generation.

Figure 4

3.2.4. Shortcomings:

One limitation of the model is that it “may experience video glitches due to network environment”. Additionally, the amount of data used for the model is limited, which may affect the model's performance. The model “may also require further refinement to improve its ability to generate diverse dance styles and movements”.

3.3. [3] J. Wen, X. Li, J. She, S. Park, and M. Cheung, Visual Background Recommendation for Dance Performances Using Dancer-Shared Images vis 2016 IEEE International Conference

3.3.1. Summary:

The model discussed is a “recommendation system that recommends backgrounds for dances” based on individual dancers' interests and dance styles. The model performs a “prediction based on Pinterest posts and uses image processing to extract only the background from top K images”. The model “provides multiple images as recommendations, taking into account the individual dancer's interests”. However, the model does not focus solely on dance style for the background and may not be suitable for large-scale applications such as movie scenes and performances.

3.3.2. Methodology:

The model uses a “prediction method based on Pinterest posts” to “recommend backgrounds for dances based on individual dancer's interests and dance styles”. The model “performs image processing to extract only the background from top K images and presents multiple images as recommendations”. The model uses a “combination of user preferences and image features to provide recommendations”.

3.3.3. Evaluation:

The model's performance is evaluated based on the “relevance and diversity of the recommended backgrounds”. The model takes “individual dancer's interests into account

and provides multiple images as recommendations". However, the model does not focus purely on dance style for background, which may affect its usefulness in some scenarios. The model "may require further refinement to improve the relevance and diversity of the recommended backgrounds".

3.3.4. Shortcomings:

One limitation of the model is that it does not focus "solely on dance style for the background", which may not be suitable for some use cases. Additionally, the model may require further refinement "to improve the relevance and diversity of the recommended backgrounds". The model may not be suitable for "large-scale applications" such as "movie scenes and performances", as it may not meet the specific requirements of those scenarios.

3.4. [4] Ho Yin Au, Jie Chen, Junkun Jiang, Yike Guo, ChoreoGraph: Music-conditioned Automatic Dance Choreography over a Style and Tempo Consistent Dynamic Graph Jul 2022

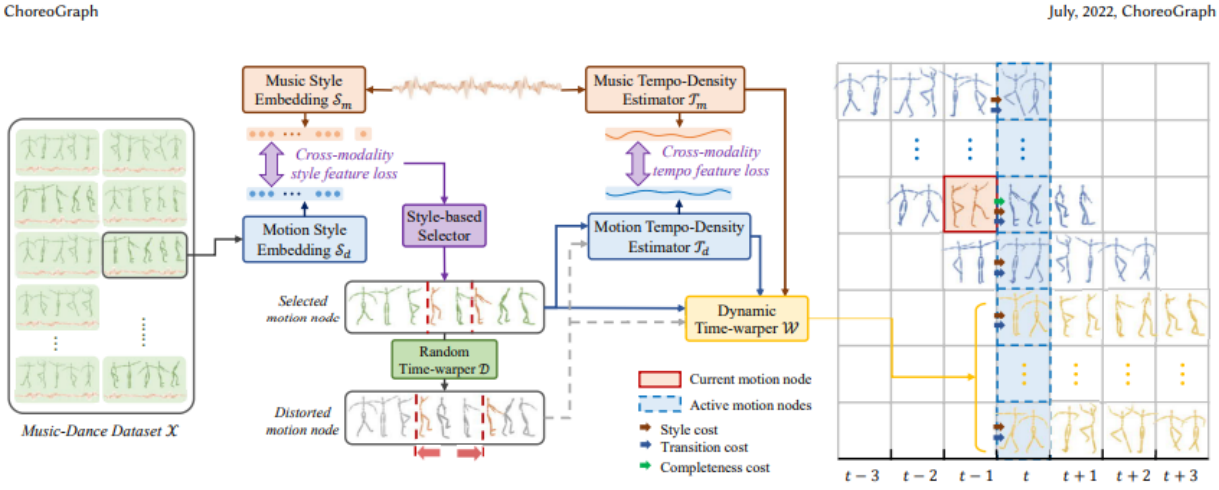

Figure 1: Proposed system diagram of our motion generation model.

Figure 5

3.4.1. Summary:

Choreographing dance that matches the music both “aesthetically and rhythmically” is a challenging task that requires taking into consideration factors such as “consistency in style and message, alignment with the musical features, and adherence to basic choreomusical rules”. To address these challenges, a framework called “ChoreoGraph” has been proposed, which generates “high-quality dance motion for a given piece of music by selecting and warping dance clips over a dynamic graph”. The framework uses a data-driven learning strategy to evaluate the “style and rhythmic connections between music and motion and generate beats-aligned motion nodes” for the graph.

3.4.2. Methodology:

“ChoreoGraph” is composed of a “tempo-density module” and a “style embedding module” for “motion selection and warping to produce motion candidate nodes”. The motion sequences are “beats-aligned” based on the music segments and then incorporated as nodes of a “dynamic motion graph”. Compatibility factors such as “style and tempo consistency, motion context connection, action completeness, and transition smoothness” are evaluated to determine the node transition in the graph. The “motion/music style embedding module” and “tempo-density prediction module” are trained on music-dance data pairs that sample and “time-warp motion sequences” into graph nodes based on extracted music style and tempo features.

3.4.3. Evaluation:

The realism of generated motions and alignment with music beats were evaluated using “Fréchet pose distance (FPD)” and “Fréchet movement distance (FMD)” and the “Beat Alignment Score (BAS)”, respectively. The results show that “ChoreoGraph” outperformed other competing methods in terms of “motion realism and motion beat alignment”, especially in the “motion2audio BAS”. However, the linear blending used in smoothing the motion node transition may affect beat alignment performance, leading to a

lower scores in “music2motion BAS”. Furthermore, “Transflower performed better in the audio2motion BAS score”, generating motion based on every music window, while “ChoreoGraph” imposes constraints on the distortion that can be applied to the content motion.

3.4.4. Shortcomings:

While “ChoreoGraph” demonstrated impressive “motion quality and diversity”, it still has some limitations. For instance, the graph-based approach may not capture the entire range of human motion, leading to less natural dance motion generation. Additionally, the motion clips used for “graph node selection” are limited by the available database, which may limit the diversity of generated dance motion. Moreover, the “linear blending method” used to smooth the motion node transition may affect “beat alignment performance”, leading to lower scores in “music2motion BAS”.

3.4.5. Dataset:

"AIST++" is a dataset that uses "(M,X)" pairs to generate dance motions that match the music both aesthetically and rhythmically. The dataset encodes "4-second music segments" and "8-second dance motions," which is used as the "database X" for the proposed model. A "style-based selector" is used to pick "K motion clips" from the dataset based on the music. The "dynamic time warper" is used to extract "4-second motions" that are "beat-aligned and tempo-adjusted" from the chosen "8-second motion clips" using the "tempo densities" of music and motion.

The motion and music are represented as "nodes in graphs," which are "1 sec long" and "organized" into "four new motion nodes" (1 sec each) into the "dynamic graph." The graph selects "one node per second" based on various factors, including "action completeness cost," "style cost," and "motion transition cost." The "chosen nodes" are then "blended into one final output motion." The dance motions are represented as "8-second

embeddings," while the music is represented as "4-second embeddings." The motion is "warped" to match the music, and the dance is represented as a "sequence of poses" (sampled at 20 frames per second), while the music is represented as a "sequence of spectral features" (sampled at 60 frames per second).

3.5. [5] Li Siyao, Weijiang Yu, Tianpei Gu, Chunze Lin, Quan Wang, Chen Qian, Chen Change Loy, Ziwei Liu, Bailando: 3D Dance Generation by Actor-Critic GPT with Choreographic Memory Mar 2022

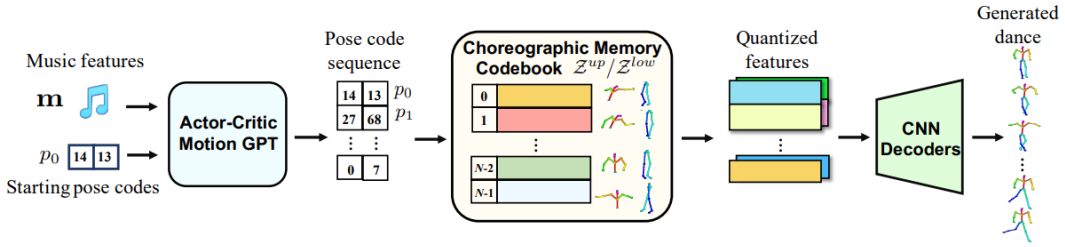

Figure 2: **Dance generation pipeline of *Bailando*.** Given a piece of music, an actor-critic motion GPT autoregressively predicts the future upper-lower pose code pairs according to the music features and starting pose codes. The pose code sequence is then embedded to quantized features via a learned choreographic memory and finally decoded into a dance sequence by a CNN-based decoder.

Figure 6

3.5.1. Summary:

"Bailando" is a new approach for generating 3D dance sequences that meet high choreography standards and are synchronized with different music genres. To tackle the challenges of dance generation, "Bailando" employs a choreographic memory that encodes and quantizes dancing-style poses, and an "Actor-critic Generative Pre-trained Transformer" ("GPT") that translates motion tempos and music beats. "Bailando" outperforms previous methods and has potential for real-world applications. Reinforcement learning scheme to the "GPT" with a newly designed beat-align reward function.

3.5.2. Methodology:

"Bailando" consists of two main components. The first component creates a choreographic memory using "VQ-VAE" to summarize "reusable components" of dancing

movements. The memory is designed to “represent both upper and lower compositional halves of 3D poses”. The second component is an “actor-critic GPT” that translates “music and source pose codes” to future pose codes. The “GPT” is optimized using “Adam optimizer” and trained with cross-conditional causal attention and beat-align reward functions with a “CNN decoder in the last step”.

.

3.5.3. Evaluation:

“Bailando” achieves its best performance on automatic metrics and visualization judgments. Quality is measured using “Frechet Inception Distances (FID)”, and diversity is computed by the average feature distance of generated movements. “Bailando” generates high-quality dance sequences that are “visually expressive and emotionally engaging”. The learned choreographic memory can discover “human-interpretable dancing-style poses without supervision”.

3.5.4. Shortcomings:

The proposed framework is computationally intensive and requires significant manual effort to collect real dancing clips as dance units. “Bailando” can still generate nonstandard poses beyond the dancing subspace, leading to unstable performance. Another limitation is that the framework is currently limited to generating 3D dance sequences and does not incorporate other aspects of dance, such as facial expressions or clothing. There is still room for improvement in terms of generating more diverse and creative dance sequences.

3.6 [6] Jonathan Tseng, Rodrigo Castellon, C. Karen Liu Stanford University, EDGE: Editable Dance Generation From Music, Nov 2022

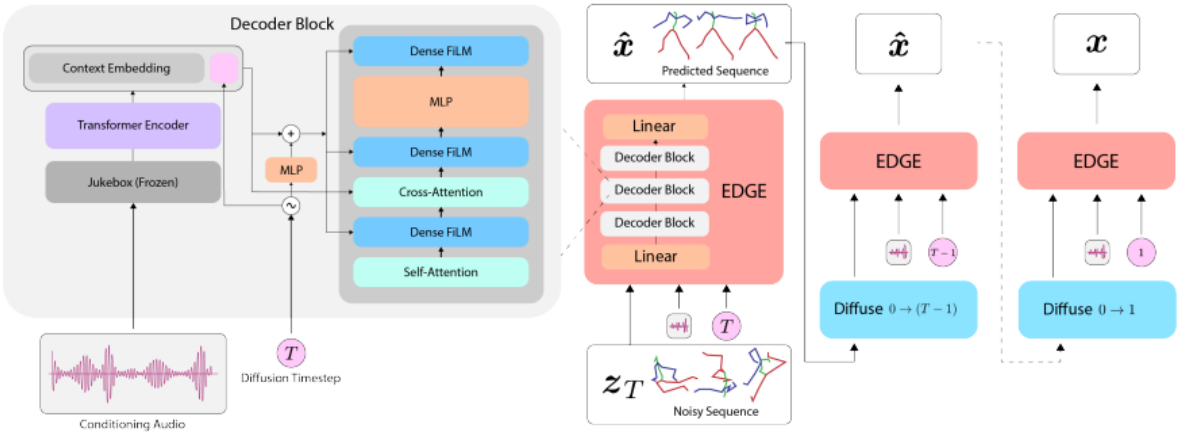

Figure 2. **EDGE Pipeline Overview:** EDGE learns to denoise dance sequences from time $t = T$ to $t = 0$, conditioned on music. Music embedding information is provided by a frozen Jukebox model [5] and acts as cross-attention context. EDGE takes a noisy sequence $\boldsymbol{z}_T \sim \mathcal{N}(0, \boldsymbol{I})$ and produces the estimated final sequence $\hat{\boldsymbol{x}}$, noising it back to $\hat{\boldsymbol{z}}_{T-1}$ and repeating until $t = 0$.

Figure 7

3.6.1. Summary:

The “Transformer-based diffusion model” used in EDGE is a state-of-the-art architecture designed to generate realistic dance movements. It uses the “Transformer decoder” and “cross-attention mechanism” to generate high-quality dance sequences. The “Jukebox model encodes input music into embeddings, which are then used to generate dance clips”. The model ensures “temporal continuity between batches of sequences, generates lower body movements given upper body movements”, and uses “prespecified motion to start and end dances with specific movements”. Additionally, the model uses “loss functions to ensure physical plausibility and visual realism”.

3.6.2. Methodology:

The methodology of “EDGE” consists of several key components. First, the Jukebox model is used to encode the input music into embeddings. These embeddings are then used as inputs to the “conditional diffusion model”, which generates a series of 5-second dance clips. The model incorporates prespecified motion, which ensures that the generated dance clips start and end with specific movements.

The pose representation used in the model is the 24-joint “SMPL format”, which uses a 6-degree of freedom rotation representation. This allows the model to generate complex and realistic dance movements.

3.6.3. Evaluation:

“EDGE” uses new and improved loss functions to ensure physical plausibility and visual realism. The “Contact Consistency Loss” prevents foot sliding, while the “Physical Foot Contact score” evaluates the amount of physical contact between the dancer's feet and the ground. These loss functions help to ensure that the generated dance clips are physically plausible and visually realistic.

The model is trained using the “AIST++ dataset”, which contains training examples that are “5 seconds in duration and captured at 30 FPS”. This allows the model to learn from a “large and diverse set of examples, which helps to improve its accuracy and fidelity”.

3.6.4. **Shortcomings**:

While the “Transformer-based diffusion model” is a highly sophisticated architecture, it still has some shortcomings. One limitation of the model is that it requires a large amount of computational resources to generate high-quality dance clips. Additionally, the model may struggle to generate dances that are “highly specific or nuanced”, as it is trained on a dataset that contains a broad range of dance styles and movements.

Another potential limitation of the model is that “EDGE” relies heavily on the input music to generate dance clips. If the input music is not well-suited to dancing or lacks a clear beat, the model may struggle to generate high-quality dance clips. Despite these limitations, the “EDGE” represents a significant advance in the field of dance generation and has the potential to be used in a wide range of applications, from “entertainment to rehabilitation”.

3.7. [7] Jinwoo Kim, Heeseok Oh2, Seongjean Kim1, Hoseok Tong1 and Sanghoon Lee*1 A Brand New Dance Partner: Music-Conditioned Pluralistic Dancing Controlled by Multiple Dance Genres, 2022

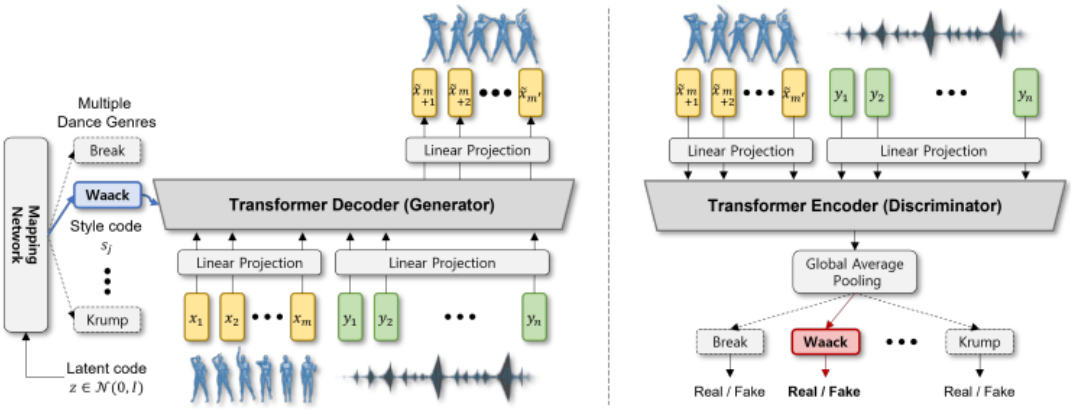

Figure 2. **Method overview:** We illustrate the generator (left) and the discriminator (right) of our transformer-based conditional GAN to generate diverse dance motions synthesized by multiple dance genres. The mapping network transforms a latent code into style code for multiple dance genres. The generator outputs long-range future motion by taking both sequences of seed motion and music piece as query and the style code as key and value. Given a sequence of real (fake) motions with a corresponding music piece, the discriminator distinguishes between real and fake motions from multiple domains.

Figure 8

3.7.1. Summary:

The “Conditional Transformer GAN” is a model designed to generate multiple genres of dance movements based on input music and seed motion. The model uses a “present mapping network to generate a style code for each dance genre and a transformer decoder to generate long-range future motion”. The transformer encoder is used as a” multi-task discriminator to distinguish between real and fake motions from multiple domains”. The model is trained on the “AIST++ dataset” and evaluated using “adversarial loss, appearance matching loss, and style diversity loss”.

3.7.2. Methodology:

The “Conditional Transformer GAN” uses a present mapping network to generate a style code for each dance genre. This code is then used as an input to the “transformer

decoder", which takes in sequences of seed motion and music pieces and generates long-range future motion. The "transformer encoder" is used as a "multi-task discriminator" to distinguish between real and fake motions from multiple domains.

The model is trained on the "AIST++ dataset", which contains a broad range of dance styles and movements. This allows the model to learn from a diverse set of examples and "generate multiple genres of dance movements with high accuracy and fidelity".

3.7.3. Evaluation:

The "Conditional Transformer GAN" is evaluated using "adversarial loss", "appearance matching loss", and "style diversity loss". "Adversarial loss" measures the ability of the discriminator to distinguish "between real and fake motions" from multiple domains. "Appearance matching loss" measures the similarity between the generated motion and the input seed motion, while "style diversity loss" measures the diversity of styles generated by the model.

The results of the evaluation show that the model is capable of generating multiple genres of dance movements with "high accuracy and fidelity". The "adversarial loss" is low, indicating that the "discriminator" is effective in distinguishing "between real and fake" motions. The "appearance matching loss" is also low, indicating that the generated motions are similar to the input seed motion. Finally, the "style diversity loss" is high, indicating that the model is capable of "generating diverse styles of dance movements".

3.7.4. Shortcomings:

While the "Conditional Transformer GAN" is a powerful model, it still has some limitations. One limitation is that it requires a large amount of computational resources to generate high-quality dance movements. Additionally, the model may struggle to generate "highly specific or nuanced dance movements", as it is trained on a dataset that contains a broad range of dance styles and movements. Another potential limitation of the model is

that it may struggle to generate realistic dance movements if the “input music is not well-suited to dancing or lacks a clear beat”.

CHAPTER-4

PROJECT REQUIREMENTS SPECIFICATION

4.1. Introduction

The primary goal of this project is to streamline the experience of choreography for dancers and choreographers alike. Using cutting-edge technologies such as Deep Learning (DL) and Language Models (LLMs), our system transforms user-selected songs into mesmerizing dance performances. By incorporating emotion and genre classification, dance movement generation, and intelligent costume and background recommendations, we aim to offer a comprehensive solution for dance enthusiasts.

4.1.1. Project Scope

Our focus extends beyond the conventional boundaries of dance creation. The system employs an Emotion Classifier to discern the nuanced emotional tones embedded within each song, mapping them to a rich palette of predefined dance styles. Simultaneously, a Genre Classifier identifies the musical genre, influencing subsequent stages of costume and background generation. The synergy between these components, guided by LLMs, culminates in a seamless fusion of auditory and visual elements, enriching the dance creation process.

4.2. Product Perspective

Dancing is a universal language found in all cultures around the world. Often, choreographing a dance for an event platforms many ambiguities. Given a certain song, classifying it into a particular genre and choosing appropriate steps given the

platforms can be a challenging task. Dancers everywhere spend much of their time, money, and effort seeking professionals to plan out their dance's choreography.

As dancers ourselves, we realize how big of a task it is to choreograph steps for different songs. Given a time constraint, it would be difficult to come up with original steps suiting the rhythm and emotion of a song. These problems led us to ideate on our AI Dance Choreographer which aims to simplify the work of countless dancers and choreographers around the world by helping them out with steps generated in a visually appealing manner in the form of a 3D human figure.

4.2.1. Product Features

1. Song Analysis and Genre Classification: Automatically analyzes uploaded songs to determine their genre. Employs an efficient genre classification system for accurate genre identification.
2. Emotion Recognition and Mapping: Utilizes an emotion recognition algorithm to identify the emotional tone of each song. Maps identified emotions to corresponding dance styles and seed motions.
3. Dance Movement Generation: Generates dynamic dance movements based on identified genre, emotion, and predefined dance styles. Utilizes seed motions to ensure expressive and stylistically appropriate choreography.
4. 3D Human Figure Visualization: Presents the generated dance choreography through a visually appealing 3D human figure. Provides an immersive and realistic representation of the dance moves.
5. Costume Recommendation and Generation: Recommends costumes based on both the genre and emotion of the song. Utilizes Language Models (LLMs) for creative and suitable costume suggestions.

6. Background Generation: Creates visually appealing backgrounds that complement the dance style and emotion of the song. Utilizes stable diffusion and LLMs to enhance the overall visual experience.
7. User-Friendly Interface: Features an intuitive and user-friendly interface for seamless song upload and interaction. Allows users to effortlessly explore and customize dance, costume, and background options.
8. Recommendation System: Provides initial suggestions for dance choreography, costumes, and backgrounds. Enables users to request alternative recommendations or choose from the provided options.

4.2.2. Operating Environment

1. Hardware platform: GPUs or TPUs are required to accelerate the computation involved in training and running the audio and dance generation models. This can be done on Google Colab.
2. Operating system and versions: The operating system used to run the system can impact its performance, security, and compatibility with other software components. If the training is done on the local system, then the OS plays an important role in the time required to train the model.
3. Software components: The software components used to develop, train, and run the deep learning model can impact its performance, accuracy, and reliability. This might include deep learning frameworks such as TensorFlow or PyTorch, audio processing libraries such as LibROSA or Essentia, or 3D rendering software such as Blender, Unity, or Vedo to generate the dance videos.
4. Development environment: The development environment used to create the system can impact the efficiency and effectiveness of the development process. This might include text editors, IDEs, version control tools, and other

software tools. We will primarily use Google Colab for the development process since it includes TPUs and GPUs as well.

5. Deployment environment: The environment in which the system is deployed can impact its availability, scalability, and reliability. This might include cloud-based infrastructure such as Amazon Web Services (AWS) or Microsoft Azure, or on-premises infrastructure such as servers or clusters. A Docker container can also be created to deploy the product to ensure portability between different systems and operating systems.

4.2.3. General Constraints, Assumptions and Dependencies

1. Data Quality Sensitivity: The accuracy of emotion and genre classification is sensitive to the quality and clarity of input songs.
2. Resource Diversity Dependency: The success of the dance movement generation relies on a diverse and comprehensive database of dance styles.
3. Flexible Visual Elements: Costume and background generation assumes flexibility to cater to diverse user preferences for customization.
4. Rendering Realism Constraints: The visual appeal and realism of 3D human figures and backgrounds are subject to the capabilities of the rendering engine.
5. User Input Reliability: Generated choreography accuracy is contingent on users providing precise emotion and genre information for the input song.
6. Database and Stylistic Standards: Assumes access to a well-curated song database and standardized dance styles mapped to emotions for effective generation.
7. Technological Dependencies: The success of the project depends on accurate machine learning models, a robust rendering engine, and external APIs for stable diffusion and creative costume suggestions.

4.2.4 Risks

1. Data Quality and Diversity: Limited availability of diverse and high-quality song and dance style data may compromise the effectiveness of emotion and genre classification.
2. Accuracy of Dance Movement Generation: Inaccuracies in dance movement generation may lead to choreography that does not align well with the emotion and genre of the song.
3. User Interface Complexity: A complex or unintuitive user interface may hinder user interaction and adoption.
4. Rendering Engine Performance: The rendering engine may struggle to produce realistic and visually appealing 3D figures and backgrounds.
5. Dependency on External APIs: Unreliable external APIs for stable diffusion and costume suggestions may disrupt the overall functionality.
6. Ethical Concerns and Bias: The AI models may inadvertently introduce biases in dance styles or recommendations.
7. Limited User Adoption: Users may not adopt the AI Dance Choreographer due to skepticism, lack of awareness, or dissatisfaction with the generated choreography.

4.3. Functional Requirements

1. Song Analysis and Genre Classification: The system must be able to analyze uploaded songs to determine their genre accurately.
2. Emotion Recognition and Mapping: Implement an emotion recognition algorithm to identify the emotional tone of each song. The map identified emotions to corresponding dance styles and seed motions.
3. Dance Movement Generation: Develop a module that generates dynamic dance movements based on identified genre, emotion, and predefined dance

styles. Utilize seed motions to ensure expressive and stylistically appropriate choreography.

4. 3D Human Figure Visualization: Present the generated dance choreography through a visually appealing 3D human figure. Ensure an immersive and realistic representation of the dance moves.
5. Costume Recommendation and Generation: Implement a system that recommends costumes based on both the genre and emotion of the song. Utilize Language Models (LLMs) for creative and suitable costume suggestions.
6. Background Generation: Develop a module that creates visually appealing backgrounds that complement the dance style and emotion of the song.Use stable diffusion and LLMs to enhance the overall visual experience.
7. User-Friendly Interface: Design an intuitive and user-friendly interface for seamless song upload and interaction. Allow users to effortlessly explore and customize dance, costume, and background options.
8. Recommendation System: Provide initial suggestions for dance choreography, costumes, and backgrounds. Enable users to request alternative recommendations or choose from the provided options.
9. Dynamic Editing and Customization: Allow users to dynamically edit and customize dance sequences. Provide flexibility for users to modify dance moves based on personal preferences.
10. Performance Analysis and Feedback: Implement a system that offers insights into the performance, including metrics on dance style consistency and emotional alignment. Facilitate continuous improvement by providing constructive feedback on choreography.

4.4. External Interface Requirements

4.4.1. User Interfaces

The user interface for this project is designed to be intuitive and visually appealing, featuring straightforward song upload mechanisms, user-friendly genre and emotion selection options, and dance style customization tools. The interface also includes sections for costume and background preferences, clear displays of AI-generated recommendations, easy-to-use editing tools for dynamic customization, performance metrics and feedback, budget estimation breakdowns, export and sharing options, a dedicated help and tutorial section, and a user profile area for revisiting and editing previous dance creations. Regular user testing and feedback collection are prioritized throughout development to ensure a seamless and enjoyable user experience.

4.4.2. Hardware Requirements

Overall, the interface linking the software and hardware components of the AI choreographer has been designed to be efficient and reliable. It is designed to support a wide range of devices and protocols, hence, making it accessible to a large audience.

1. Supported Devices: The Dance Choreographer system is mainly a web-based application so it can be accessed from any device that has a web browser and an internet connection. This includes devices such as desktop computers, laptops, tablets, and smartphones.
2. Protocols: The interface linking the software product and the hardware components of the system supports many protocols such as HTTP/HTTPS, TCP/IP, and SSL/TLS. These protocols are used for secure communication between the client and server interfaces.

3. Communication: The interface linking the software product and the hardware components of our system are based on a client-server architecture. The client-side interface (i.e., the web browser) communicates with the server-side interface (i.e., the web application) using the HTTP/HTTPS protocol.
4. CPU and Memory Requirements: The AI Dance Choreographer system requires a minimum CPU and memory capacity to function efficiently. The requirements for the CPU are 1.8 GHz or higher, and the memory requirement is 4GB RAM or higher.
5. Display and Resolution: For optimal user experience, the AI Dance Choreographer system needs a display with a minimum resolution of 1366x768 pixels.
6. Input and Output Devices: The AI Dance Choreographer system requires input devices such as a keyboard and mouse for user input. Speakers would also be necessary to play the music audio.
7. Internet Connection: The AI Dance Choreographer system requires a stable internet connection with a minimum speed of 10 Mbps or higher.

4.4.3. Software Requirements

Product: Automated Dance Choreography and Direction Using Music Classification and Contextual Factors

Description: The AI Dance Choreographer/Director system is an AI-powered web application that generates customized dance choreographies based on user inputs such as dance styles, audio or video recordings, and user preferences.

Version/Release Number: 1.0

Operating Systems:

1. Windows 10 or later

2. MacOS 10.15 or later
3. Ubuntu 20.04 or later

Tools and Libraries:

1. Python 3.8 or later: Python is used as the primary programming language to develop AI models and algorithms.
2. TensorFlow 2.4 or later: TensorFlow is used to develop machine learning models for generating choreographies.
3. OpenCV 4.5 or later: OpenCV is used for image and video processing.

Any changes or modifications made to these software components may impact the system's performance and functionality. It is, therefore, important to ensure that these software components are up-to-date and compatible with each other. Additionally, regular software updates and maintenance are necessary to ensure the system's security, stability, and performance.

4.4.4. Communication Interfaces

The system shall be compatible with different devices, operating systems, and browsers. It should support popular web browsers, mobile devices, and operating systems such as Windows, MacOS, iOS, and Android. The system requires a high-speed internet connection to process input data, generate choreographies, and provide visual feedback to users and requires a sufficient buffer size to store input data and generated choreographies temporarily. The system should use encryption protocols such as Secure Socket Layer (SSL) or Transport Layer Security (TLS) to protect user data and ensure secure communication between the user and the system. The system should comply with API standards to enable seamless integration with other systems or platforms.

4.5. Non-Functional Requirements

1. Accuracy and precision: The AI dance choreographer should be able to accurately and precisely detect, analyze, and create dance moves. It should also be able to create choreographies that match the music, rhythm, and beat of the selected song.
2. Real-time performance: The AI dance choreographer should be able to respond in real time to changes in the music, tempo, or other environmental factors. It should be able to adapt to changes and continue generating appropriate dance moves.
3. Scalability and performance: The AI dance choreographer should be able to handle a large volume of data and process it efficiently. It should also be able to handle multiple dance styles and genres.
4. User interface and ease of use: The AI dance choreographer should have a user-friendly interface that is easy to navigate and use. It should also have clear instructions for users on how to use it.
5. Robustness and reliability: The AI dance choreographer should be able to handle errors, unexpected inputs, and exceptions gracefully. It should also be reliable and consistent in its performance.
6. Security and privacy: The AI dance choreographer should be designed with security and privacy in mind, to protect the user's data and prevent unauthorized access.
7. Compatibility and integration: The AI dance choreographer should be able to integrate with other systems, platforms and tools seamlessly. It should also be compatible with different devices, operating systems and browsers.

4.5.1. Performance Requirement

1. Song Analysis Speed: The system should analyze and classify the genre and emotion of a song within a reasonable time frame, aiming for real-time or near-real-time responsiveness.
2. Dance Movement Generation Time: The generation of dance movements, including seed motions and choreography, should occur efficiently, providing users with prompt results to maintain engagement.
3. Rendering Engine Speed: The rendering of 3D human figures and backgrounds should be smooth and responsive, ensuring a visually appealing and immersive experience for users.
4. User Interface Responsiveness: User interactions with the interface, including song uploads, genre/emotion selection, and customization, should be responsive and free from noticeable delays.
5. Recommendation System Latency: The system should swiftly generate and present AI-generated recommendations for dance choreography, costumes, and backgrounds, minimizing wait times for users.
6. Editing and Customization Efficiency: The editing tools for customizing dance sequences should respond promptly, allowing users to make dynamic adjustments with minimal latency.
7. Performance Metrics Computation: The computation of performance metrics, such as dance style consistency and emotional alignment, should be efficient and not introduce delays in the user experience.
8. Scalability and Concurrent Users: The system should be scalable to accommodate multiple concurrent users without significant degradation in performance.
9. Resource Utilization: The project should utilize computational resources efficiently to prevent excessive consumption of processing power and memory.

10. Reliability and Uptime: The system should maintain a high level of reliability, minimizing downtime and ensuring users can access the application consistently.

4.5.2. Safety Requirements

1. Risk Assessment: The AI Dance Choreographer system should undergo a comprehensive risk assessment to identify potential hazards or risks associated with generated dance movements, aiming to ensure user safety and minimize the risk of injury.
2. User Skill Level Assessment: The system must be capable of assessing the user's dance skill level accurately, enabling the generation of choreography that aligns with their experience and ability, promoting a safe and enjoyable dance experience.
3. Safety Guidelines: Clear and concise safety guidelines should be provided within the system, including information on proper warm-up and cool-down procedures. Users must be informed about potential risks associated with specific dance movements to enhance their awareness and safety.
4. Progression Tracking: The system should track user progress systematically and dynamically adjust the difficulty level of generated choreography. This adaptive feature aims to prevent injuries and ensure a safe and effective dance workout experience over time.
5. User Feedback on Safety: Users should have the capability to provide feedback on the quality and safety of the generated choreography, including costume and background suggestions. This feedback loop is essential to identify potential safety issues and areas for improvement in the entire choreography generation process.
6. Personalization for Safety: The system should personalize generated choreography, including costumes and backgrounds, based on the user's fitness level, physical limitations, and other relevant factors. Users should also have the option to choose

the desired difficulty level of the dance, ensuring a customized and safe workout experience.

7. Costume and Background Safety Considerations: Safety guidelines related to costumes and backgrounds should be integrated into the system, providing information on materials, fitting considerations, and potential visual or environmental triggers to ensure user comfort and safety.
8. Emergency Protocols: The system must include clear instructions and protocols for users to follow in case of an emergency, such as injury or illness during a dance session. Appropriate measures should be in place to respond promptly and effectively to emergency situations.

4.5.3. Security Requirements

1. Data protection: The system should be designed to protect user data, such as personal information and login credentials, using encryption and other security measures.
2. Access control: The system should have appropriate access controls in place to ensure that only authorized users are able to access the system and perform certain actions.
3. Authentication and authorization: The system should use strong authentication and authorization protocols to verify the identity of users and ensure that they have appropriate permissions to access and use the system.
4. Secure communication: The system should use secure communication protocols, such as SSL/TLS, to ensure that data is transmitted securely between the system and users.
5. Audit logs: The system should maintain audit logs of all user activity and system events, in order to detect and respond to any potential security incidents.

6. Vulnerability management: The system should have appropriate vulnerability management processes in place to identify and address any potential security vulnerabilities or weaknesses.
7. Regular security assessments: The system should undergo regular security assessments, such as penetration testing, to identify any potential security issues and ensure that the system remains secure over time.

Appendix A: Definitions, Acronyms and Abbreviations

List of Acronyms:

- **GPU** Graphics Processing Unit
- **TPU** Tensor Processing Unit
- **IDEs** Integrated development environments
- **UI** User Interface
- **API** Application Programming Interface
- **SSL** Secure Socket Layer
- **TLS** Transport Layer Security

Appendix B: References, Use Cases

Use Cases:

1. Dance Creation: The AI dance choreographer can generate dance routines and sequences based on user inputs, such as preferred dance style, tempo, and mood. Users can also provide feedback on the generated routines, allowing the AI to learn and improve its dance creation algorithms.

2. Personalization: The AI dance choreographer can personalize dance routines for individual users based on their physical abilities and preferences. The AI can analyze data from sensors, such as motion capture systems or wearable devices, to adjust the difficulty level of the dance routine.
3. Learning: The AI dance choreographer can be used as a tool for dance education and training. The AI can provide feedback on a user's movements and suggest corrections or improvements to their technique. It can also generate customized practice routines for users based on their skill level and learning goals.
4. Performance: The AI can generate choreography that is synchronized with music and lighting cues, ensuring a seamless and visually stunning performance.
5. Entertainment: The AI dance choreographer can be used as a tool for entertainment, allowing users to create and share dance routines with their friends and family. The AI can also generate dance routines that are tailored to specific events, such as weddings, parties, or corporate events.

CHAPTER-5

SYSTEM DESIGN

5.1 Introduction

The purpose of this document is to provide a high-level design of an Automated Dance Choreography and Direction System that uses music classification and contextual factors. The system aims to classify music into genres and map it to a dance style using a model trained on AIST dataset videos, and then generate visually appealing background settings for the dance to be performed. The system is designed to have applications in the entertainment industry as well as education, such as in dance classes or workshops.

5.2 Current System

In traditional dance choreography, the choreographer manually creates the dance moves and sequences them to match the music. However, with the proposed system, dance moves are generated algorithmically based on the genre and emotion of the music, allowing for a more automated and efficient process. The system could potentially revolutionize the entertainment industry by providing a faster and more cost-effective way to generate dance performances and music videos.

5.3 Design Considerations

5.3.1 Design Goals

The design goals of the proposed system are to provide a more efficient and effective method of generating dance choreography and direction using music classification and contextual factors. The system should be user-friendly and provide a visually appealing experience for the end-users. The system should also

be reliable, secure, and scalable.

5.3.2 Architecture Choices

The proposed system will use the FACT transformer modified with music classification using emotion with the Spotipy library and visualized appropriately with a suitable background. For the purpose of generating a background for the dance and a costume for the dancer, a large language model (LLM) is used to create a prompt which is fed to a stable diffusion model. These choices were made because they are widely used and have been proven to be effective in similar projects.

5.3.3 Constraints, Assumptions, and Dependencies

Assumptions and Dependencies:

1. The input music will be of high quality and suitable for dance choreography.
2. The system depends on the quality and suitability of the input music for dance choreography. Low-quality music may lead to inaccurate music classification and mapping, resulting in unsuitable dance motions.
3. End-users will have a basic understanding of dance and music genres.
4. The system assumes that the end-users have some knowledge of dance and music genres to ensure that the generated dance choreography and background settings are suitable for the music and dance style.
5. A stable and reliable internet connection is available.
6. The system depends on a stable and reliable internet connection to access the Spotipy library and AIST dataset videos for music classification and dance style mapping.
7. Adequate resources are available to support the system.

8. The system requires adequate computing resources, including processing power, memory, and storage, to generate dance motions and background settings. Insufficient resources may result in poor performance or system failure.
9. The system will be used primarily in the entertainment industry and education.
10. The system is designed for use in the entertainment industry and education, and its effectiveness in other industries is not guaranteed.

Constraints:

1. Interoperability requirements The proposed system must integrate with existing systems and technologies used in the entertainment industry, such as music and video production software.
2. Hardware and software environment The system must be compatible with the hardware and software environment of the end users. Compatibility issues may arise if the end-users are using different operating systems or hardware configurations.
3. Performance-related issues The system must be able to generate dance motions and background settings in a timely manner to meet the demands of the entertainment industry. Poor performance may lead to delays in production and reduced customer satisfaction.
4. Deployment and scalability The system must be easily deployable and scalable to meet the changing needs of the entertainment industry. The system must also be maintainable and upgradable to keep up with technological advancements and changing industry standards.
5. Data repository and distribution requirements The system must be able to handle large amounts of data, including music and dance videos, and distribute them efficiently to end-users. The system must also ensure data privacy and security to protect sensitive information.

5.4 High-Level System Design

The high-level system design of the proposed system consists of four perspectives: conceptual, process, physical, and security.

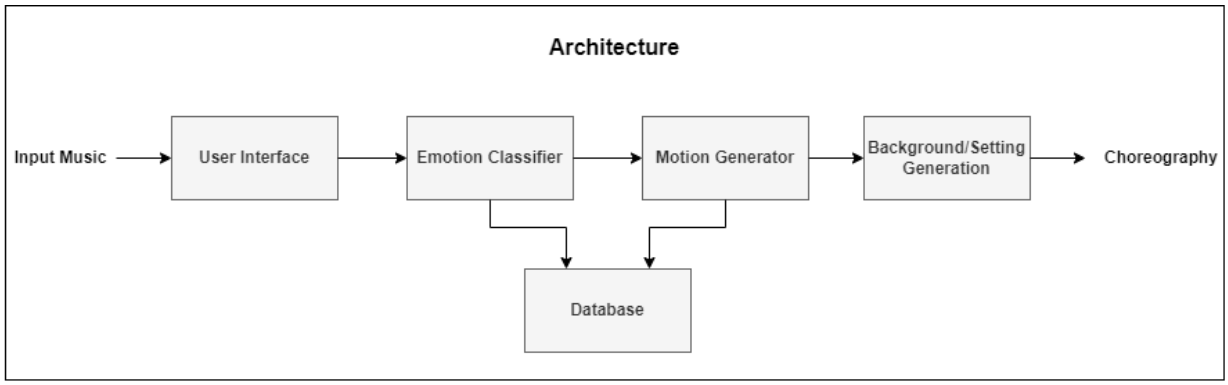

Figure 9

5.4.1 Conceptual or Logical Design

The conceptual or logical view of the system includes the following major components:

• Music classification component: This component will classify the input music into genres using the Spotipy library.

• Dance style mapping component: This component will map the classified music to a dance style using a transformer model trained on AIST dataset videos.

• Motion generation component: This component will generate dance motions based on the mapped dance style and the emotion of the music. The generated dance will be applied to a 3D mesh figure for easy visualization.

• Background setting and costume generation component: This component will generate visually appealing background settings for the dance to be performed based on the mapped dance style and the emotion of the music. Moreover, costumes

and lighting will also be added to create a visually pleasing and immersive experience.

5.4.2 Process

The process view of the system illustrates the runtime behavior of the system. The major components of the system and their interactions are shown in an interaction diagram. The major components of the system are a music feature extractor, emotion classifier, dance generator, visualizer, and setting generator. The interconnection of these components is shown in the diagram above.

5.4.3 Security

The security view of the system describes the security features of the system. The system will have secure login and authentication features to ensure that only authorized users can access the system. The dances generated for a particular user will be available only to that user through the login functionality.

5.5 Design Description

5.5.1 Master Class Diagram

• The system consists of several major modules, including the user interface module, music classification module, dance style mapping module, motion generation module, and background setting generation module. The modules interact with each other to generate dance choreography and direction based on the input music and contextual factors.

• The user interface module allows the end-users to interact with the system and input music. The music classification module classifies the input music into genres and emotions using the SpotiPy library. The dance style mapping module maps the classified music to a dance style using a machine learning model

trained on AIST dataset videos. The motion generation module generates dance motions based on the mapped dance style and the emotion of the music, and the background setting generation module generates visually appealing background settings for the dance to be performed.

5.5.2 Reusability Considerations

The following reusability considerations are planned for the project:

• Project Components that are and can be generated with available reusable components: The system will use existing libraries and frameworks such as SpotiPy and machine learning models for music classification and dance style mapping to minimize the need for custom development. This will also ensure that the system is compatible with other systems and technologies used in the entertainment industry.

• Components that can be built in the project for reuse in the project: The system will use modular design and development practices to promote reusability. For example, the motion generation module can be reused for generating dance motions for other music genres and dance styles. Similarly, the background setting generation module can be reused for generating backgrounds for other dance performances.

5.5.3 Master Class Diagram

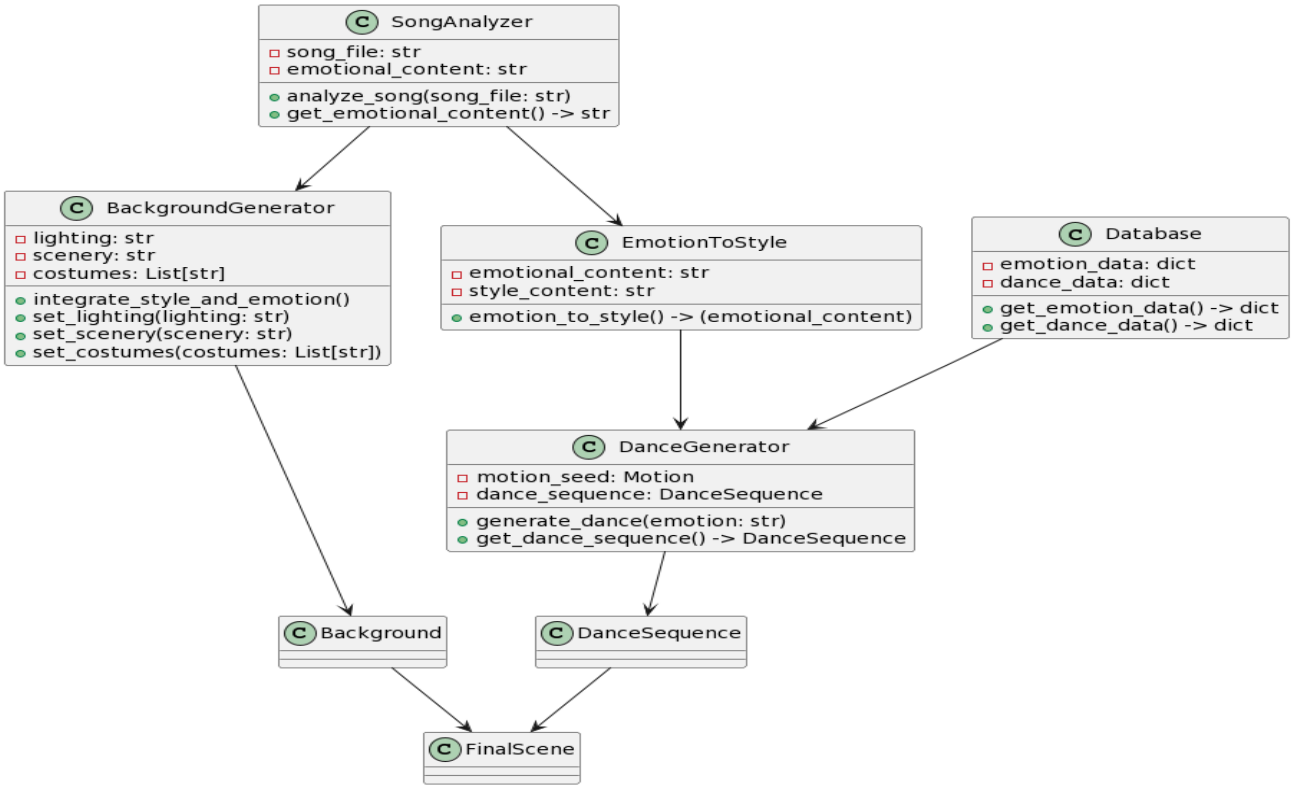

Figure 10

5.5.4 Use Case Diagram

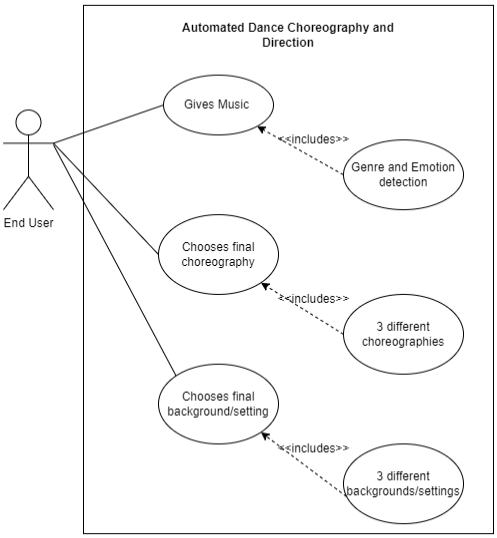

Figure 11

5.5.5 Report Layouts

The system will generate reports for the dance choreography and direction generated based on the input music and contextual factors. The report will include the following information:

• Dance style: The dance style that best suits the input music, as determined by the dance style mapping module.

• Emotion: The emotion of the input music, as determined by the music classification module.

• Dance motions: The dance motions generated by the motion generation module based on the mapped dance style and the emotion of the music.

• Background setting: The visually appealing background setting generated by the background setting generation module based on the mapped dance style and the emotion of the music.

The report can be sorted and grouped based on various criteria, such as dance style, emotion, and date of creation. The report can also be filtered based on specific criteria, such as music genre, dance style, and emotion.

The report layout will be designed to be user-friendly and easy to read, with clear headings and columns. The actual report layout will be provided in an appendix to this document, along with details on the columns used in the report.

5.6 External Interfaces

The following components will be interfaced externally by the system:

The music player that will supply the system's input music will interface with the system. Any common music player that is capable of playing files in the MP3, WAV, and AIFF formats can be used as the music player.

• Database: Information regarding the dance choreography and direction produced by the system will be stored in a database that the system will interface with. Reports will be generated and historical dance performance data will be retrieved from the database.

• Visualisation software: The dance motions and background created by the system will be rendered by visualization software, which the system will interface with. Any standard 3D animation program that supports file formats like FBX and OBJ can be used as the visualisation tool.

The following components will be interfaced externally by the system:

The music player that will supply the system's input music will interface with the system. Any common music player that is capable of playing files in the MP3, WAV, and AIFF formats can be used as the music player.

• Database: Information regarding the dance choreography and direction produced by the system will be stored in a database that the system will interface with. Reports will be generated and historical dance performance data will be retrieved from the database.

• Visualisation software: The dance motions and background created by the system will be rendered by visualisation software, which the system will interface with. Any standard 3D animation program that supports file formats like FBX and OBJ can be used as the visualisation tool.

Users can interact with the system via the GUI, and the visualisation software will render the dance movements and background created by the system.

5.7 Help

To assist with system usage, the system will provide online, context-sensitive help. User and technical manuals for the system will also be accessible for reference; these will contain comprehensive explanations of the features and functions of the system as well as step-by-step instructions on how to operate it. All technical users will be able to easily navigate and access the help and documentation.

5.8 Design Details

Numerous platforms, systems, and procedures will influence the system's design. The platforms, systems, and procedures that the system will rely on include the following examples:

1. Novelty: The system is a new approach to automated dance direction and choreography. It maps musical genres to dance styles and creates dance steps depending on the emotional content of the song using machine learning and computer vision techniques.
2. Innovation: The system automates dance choreography and direction by combining various technologies in a unique way. It offers an easy solution to make dance videos

and music videos, which has the potential to completely transform the entertainment industry.

3. Interoperability: For the system to work properly, it must be able to communicate with various external interfaces, including a music player, database, and visualization software.
4. Performance: To handle the machine learning algorithms and 3D animation rendering, the system will need to have sufficient memory and processing power. It might be necessary to use performance optimization strategies to make sure the system functions well.
5. Security: To guarantee that only authorized users, secure login and authentication features are required in the system. It will be necessary to enforce safeguards against cyberattacks and data breaches.
6. Reliability: The system has to be dependable, operating error-free, and not have any downtime. It will be necessary to execute testing and quality assurance procedures to ensure the dependability of the system.
7. Maintainability: To keep the system current and operating properly, it requires updates and maintenance. Implementing version control and code restructuring procedures is necessary to guarantee maintainability.
8. Portability: The system must be able to operate on various hardware and software environments to be a portable system. To guarantee portability, cross-platform development and deployment procedures must be applied.
9. From legacy to modernization: We offer an automated and effective method for producing dance performances and music videos, the system has potential to make the entertainment sector modern. It may take the place of the manual, conventional method of choreography and direction for dance as well.
10. Reusability: Parts that can be integrated into the project and used again in the same or different projects are included in the project.

11. Application compatibility: For the system to operate properly, it must be compatible with a variety of programs and applications. To guarantee compatibility, procedures for compatibility testing must be put in place.
12. Resource utilization: To guarantee optimum performance, the system must make effective use of its hardware and software resources. To guarantee resource efficiency, resource utilization optimisation techniques must be put into practise.

Appendix A: Definitions, Acronyms and Abbreviations

"GUI: Graphical User Interface"

"MP3: MPEG Audio Layer 3"

"WAV: Waveform Audio File Format"

"AIFF: Audio Interchange File Format"

"FBX: Filmbox"

"OBJ: Object file"

"AIST++: 3D dance dataset which contains 3D motion reconstructed from real dancers paired with music."

CHAPTER-6

IMPLEMENTATION AND PSEUDO CODE

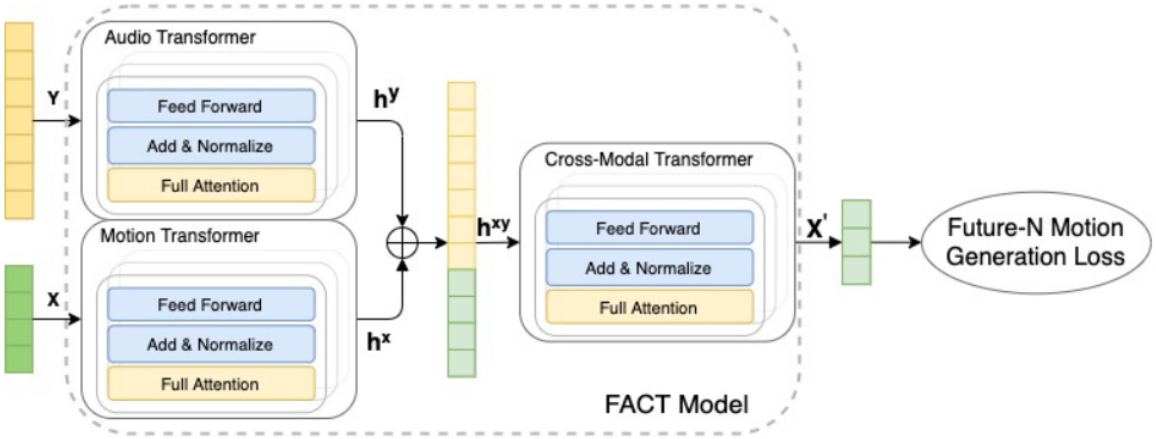

Figure 12

The above is a diagram of the Full Attention Cross-Modal Transformer which is the model we are using to generate dances. The model consists of 3 different transformers that perform different tasks. The audio transformer extracts necessary audio features from the input music such as tempo, beats, etc. The motion transformer encodes seed motion, which is a randomized starting step for any dance sequence. These audio and motion embeddings are combined and sent into yet another transformer which is the cross-modal transformer which generates the future sequences. The training happens in a self-supervised manner using these sequences. During testing, the model is applied in an auto-regressive framework

where the predicted sequence is served as input to the next step. The seed motions are chosen based on emotion of the different parts of the song, using an SVM.

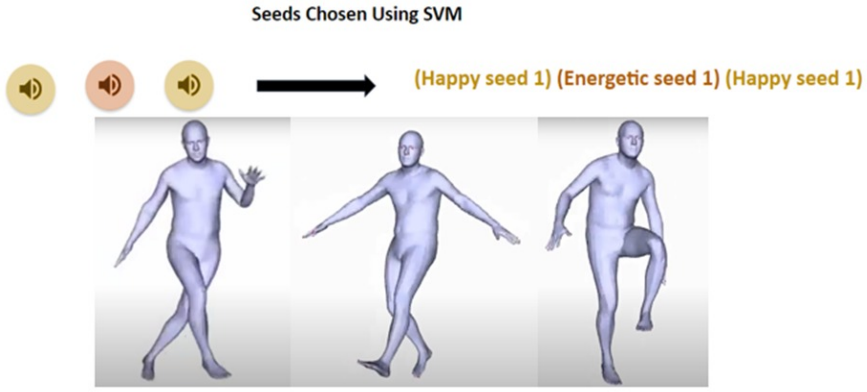

Figure 13

Emotion classification to map audio to dance style:

```
spotify = spotipy.Spotify(client_credentials_manager=client_credentials_manager)

playlists = {
            'Energetic' : ["https://open.spotify.com/playlist/4QDWboU5rwpDRXwYprwJf5?si=815ef93592e7475f",
                           "https://open.spotify.com/playlist/0V32mTwWBzo6rNIk21owsY?si=1421ae7c97434fbf"],
            'Relaxing':   ["https://open.spotify.com/playlist/1r4hnyOWexSvylLokn2hUa?si=a622dcb83906450f",
                           "https://open.spotify.com/playlist/11IcIUefRdjIpy1K5GMdOH?si=83b430249c854434"],
            'Dark':       ["https://open.spotify.com/playlist/52fjgY5XTdBKrk71TJks1i?si=28df39510a12411e",
                           "https://open.spotify.com/playlist/5xLGNk2Hqi47S1skLPIIZg?si=bb90318b09644f6a"],
            'Aggressive':  ["https://open.spotify.com/playlist/0y1bZzglw6D3t5lPXRmuYS?si=57c0db5e01b9420b",
                            "https://open.spotify.com/playlist/7rthAUYUFcbEeC8NS8Wh42?si=ab7f353f5f2847b3"],
            'Sad':         ["https://open.spotify.com/playlist/2lXXUXSsI8xt7CYxnh7r2O"],
            'Happy' :      ["https://open.spotify.com/playlist/1h90L3LP8kAJ7KGjCV2Xfd?si=5e2691af69e544a3",
                            "https://open.spotify.com/playlist/4AnAUkQNrLKlJCInZGSXRO?si=9846e647548d4026"],
}
tracks = pd.DataFrame()
moods = []
```

Figure 14

```
for i in l:
    print(i)
    if(i==1):
        no = random.randint(1,len(d["Energetic"]))
        print(d["Energetic"][no])
    elif (i==2):
        no = random.randint(1,len(d["Relaxing"]))
        print(d["Relaxing"][no])
    elif(i==3):
        no = random.randint(1,len(d["Dark"]))
        print(d["Dark"][no])
    elif (i==4):
        no = random.randint(1,len(d["Aggressive"]))
        print(d["Aggressive"][no])
    elif(i==5):
        no = random.randint(1,len(d["Sad"]))
        print(d["Sad"][no])
    elif(i==6):
        no = random.randint(1,len(d["Happy"]))
        print(d["Happy"][no])
```

Figure 15

For costume generation, our approach centers on a detailed examination of two crucial elements ingrained in music—specifically, emotion and genre. These aspects are derived through a thorough analysis of music features like beat and tempo. The costume generation process unfolds in two main phases. Firstly, a comprehensive prompt is crafted using a large language model (LLM), encompassing keywords that outline the garments and accessories suited for the dance. This prompt is carefully tailored based on the emotion and genre of the song. Subsequently, this refined prompt seamlessly integrates into a stable diffusion model designed for transforming text into images, resulting in a fitting costume. This method ensures a precise alignment between the costume and the distinctive characteristics of the music, showcasing the efficacy of our approach.

Background generation follows a similar methodology. Considering the genre of the song, an LLM is employed to formulate a prompt describing an ideal dance setting that complements the mood and tonality of the song. This prompt is then input into

a stable diffusion model, generating an image portraying a potential background that users can seamlessly incorporate into their dance routine.

Below is a figure of the UI frontend for our application incorporating the generated dance, as well as costume and background generated using stable diffusion based on the genre of the song.

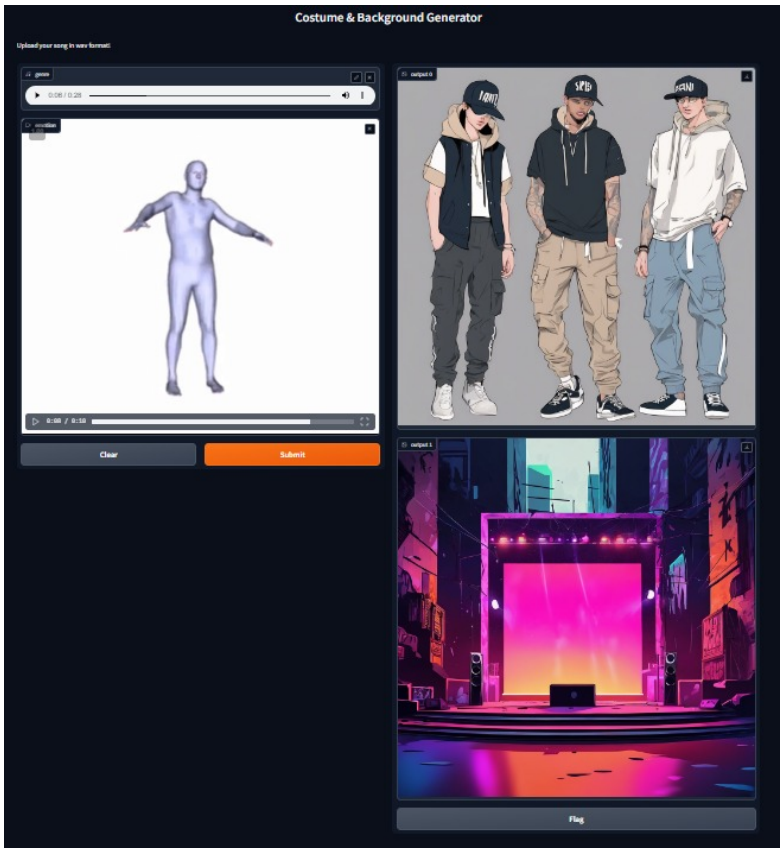

Figure 16

CHAPTER-7

RESULTS AND DISCUSSION

The accuracy for the music genre classification as produced by the KNN algorithm is calculated by dividing the count of correctly predicted instances by the total number of instances in the test set. The counter is incremented for each instance where the predicted genre label matches the actual label. Multiplying by 1.0 ensures the result is a floating-point number. The formula for the same:

Accuracy = Total Number of Instances in the Test Set / Number of Correct Predictions

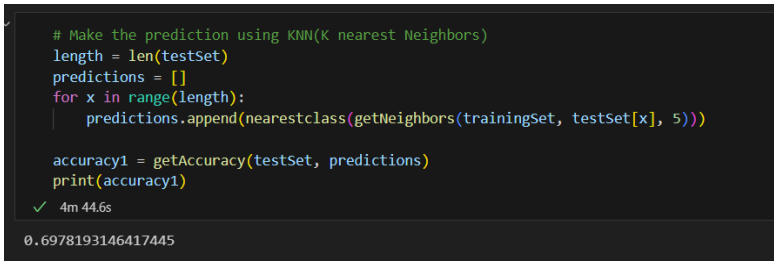

Figure 17

To measure the accuracy for mapping the seed motions, we used the Hinge Loss function is defined as:

L(y, f(x)) = max(0, 1 - y * f(x))

The Hinge Loss function uses the proportion of the distance from the SVM margin as a measure of accuracy of classification. The objective is to minimize Hinge Loss. A well-classified SVM would usually have hinge loss values close to or equal to 0 for the correct class and positive hinge loss values for the incorrect classes.

Objective = Σ L(yi, f(xi)) + λ * ||w||2

We got an average Hinge Loss of 0.57 after testing on various different song genres and emotions.

CHAPTER-8

CONCLUSION AND FUTURE WORK

To show how the model works with different kinds of songs, audio elements have been taken from a variety of sources. Using a neural network, an emotion classifier has been developed that can reliably categorise audio data into six different emotion classes: energetic, calming, dark, aggressive, sad, and happy. Moreover, SVM is employed to select the emotion-appropriate seed motion. We have created several dances for every song using the baseline FACT model in order to observe how distinctively each dance is created using a new seed motion each time. The K-Nearest Neighbours (KNN) algorithm, which is supplied as one of the parameters in the costume and backdrop production pipelines, is used to identify the genre of the song that the user has input. Providing the user with an appropriate outfit and a suitable atmosphere for the created choreography to be danced in is a major component of this project. A stable diffusion model and a large language model (LLM) are used to produce both of these results. The prompt that is produced by the LLM is fed into the diffusion model. The user will have an engaging visual experience thanks to the integrated process, which includes all of the phases mentioned above and provides a complete product that includes everything from dance generation to direction and setup.

REFERENCES

[1] R. Li, S. Yang, D. A. Ross, A. Kanazawa, AI Choreographer: Music Conditioned 3D Dance Generation with AIST++ vis ICCV 2021

[2] Liu Xin, Ko Young Chun, The use of deep learning technology in dance movement generation vis 2022 Frontiers in Neurorobotics 16

[3] J. Wen, X. Li, J. She, S. Park, and M. Cheung, Visual Background Recommendation for Dance Performances Using Dancer-Shared Images vis 2016 IEEE International Conference

[4] Ho Yin Au, Jie Chen, Junkun Jiang, Yike Guo, ChoreoGraph: Music-conditioned Automatic Dance Choreography over a Style and Tempo Consistent Dynamic Graph Jul 2022

[5] Li Siyao, Weijiang Yu, Tianpei Gu, Chunze Lin, Quan Wang, Chen Qian, Chen Change Loy, Ziwei Liu, Bailando: 3D Dance Generation by Actor-Critic GPT with Choreographic Memory Mar 2022

[6] Jonathan Tseng, Rodrigo Castellon, C. Karen Liu Stanford University, EDGE: Editable Dance Generation From Music, Nov 2022

[7] Jinwoo Kim, Heeseok Oh2, Seongjean Kim1, Hoseok Tong1 and Sanghoon Lee*1 A Brand New Dance Partner: Music-Conditioned Pluralistic Dancing Controlled by Multiple Dance Genres, 2022

APPENDIX A DEFINITIONS, ACRONYMS AND ABBREVIATIONS

List of Acronyms:

“GPU: Graphics Processing Unit”

“TPU:Tensor Processing Unit”

“IDEs: Integrated development environments”

“UI: User Interface”

“API: Application Programming Interface”

“SSL: Secure Socket Layer”

“TLS: Transport Layer Security”

GUI: Graphical User Interface”

“MP3: MPEG Audio Layer 3”

“WAV: Waveform Audio File Format”

“AIFF: Audio Interchange File Format”

“FBX: Filmbox”

“OBJ: Object file”

“AIST++: 3D dance dataset which contains 3D motion reconstructed from real dancers paired with music.”

“KNN: K-Nearest Neighbours”

“LLM: Large Language Model”

Word Count: 11557

Plagiarism Percentage 9%

Matches

1 **World Wide Web Match**
View Link

2 **World Wide Web Match**
View Link

3 **World Wide Web Match**
View Link

4 **World Wide Web Match**
View Link

5 **World Wide Web Match**
View Link

6 **World Wide Web Match**
View Link

7 **World Wide Web Match**
View Link

8 **World Wide Web Match**
View Link

9 **World Wide Web Match**
View Link

10 **World Wide Web Match**
View Link

11 **World Wide Web Match**

Notification of Acceptance of the PRML 2023

August 4-6, 2023, Urumqi, China

http://www.prml.org

Paper ID : AU0102

Paper Title : UniQNatyam: An Approach Towards Non-Repetitive Dance Generation

Dear Vibha H Murthy, Vidisha Chandra, Vishakha Hegde, Rayudu Srishti, and K S Srinivas,

First of all, thank you for your concern. 2023 IEEE the 4th International Conference on Pattern Recognition and Machine Learning (PRML 2023) review procedure has been finished. We are delighted to inform you that your manuscript has been accepted for presentation at PRML 2023 which will be held during August 4-6, 2023 in Urumqi, China. Your paper was tripling blind-reviewed and based on the evaluations. The reviewers' comments are enclosed.

The conference received papers from about 17 different countries and regions during the submission period. And there are about 149 papers accepted by our reviewers who are the international experts from all over the world. The selected papers could be published in the conference proceedings with high quality. According to the recommendations from reviewers and technical program committees, we are glad to inform you that your paper identified above have been selected for publication and oral presentation. You are invited to present your paper and studies during our PRML conference that would be held on August 4-6, 2023, Urumqi, China.

PRML 2023 is co-sponsored by Xinjiang University, Sichuan University and IEEE, supported by University of Electronic Science and Technology of China, Xi'an Jiaotong University, Tibet University, Wenzhou Medical University and Universiti Malaya. PRML 2023 has entered the IEEE upcoming conference list.

PRML 2023 papers will be published in PRML 2023 IEEE Conference Proceedings, which will be included in IEEE Xplore, indexed by Ei Compendex and Scopus.

(Important Steps for your registration): Please do finish all the 4 steps on time to guarantee the paper published in the proceeding successfully:

1. Revise your paper according to the Review Comments in the attachment carefully. (Eight authors at most each paper)

2. Format your paper according to the Template carefully.
http://prml.org/template.docx
http://prml.org/IEEE%20Latex%20Template.zip

3. Register and pay Registration fee through the online system.

http://confsys.iconf.org/register/prml2023 (An account is needed for online registration.)

4. Send your final papers (both .doc and .pdf format) and payment (in .jpg format) to us at icprml@163.com. (Before August 24, 2023) (Very important)

Address all reviewers' comments and revise your paper accordingly before submitting the final camera-ready version. Please make sure that all Figures and Tables are of high quality and their content is easily readable. Once submitted, **no revisions will be accepted**.

Conference secretary will contact you later about the copyright, please pay attention to your e-mail box.

Cancellation and Refund Policy

All cancellations must be sent in writing to the conference email box: icprml@163.com and are subject to the following cancellation charges:

*** Conference registration cancelled before May. 05, 2023 with reasonable grounds: 100% of the payment will be refunded, however an administration fee of 30USD will be charged for the service.**

*** Conference registration cancelled before Jun. 05, 2023 with reasonable grounds: 50% of the payment will be refunded.**

*** Conference registration cancelled before Jun. 25, 2023 with reasonable grounds: 30% of the payment will be refunded.**

*** Conference registration cancelled after Jun. 25, 2023 with reasonable grounds: the payment will not be refunded.**

**** 1. The above item is based on deduction for any bank transfer fees (30USD). 2. In case of visa rejection, no-show or early departure, no refund of registration fees will be made. 3. All refunds will be processed after the conference. 4. Due to force majeure including but not limited to earthquake, natural disaster, war and country policy, organizer reserves the rights to change the conference dates or venue with immediate effect and does not assume responsibility for any additional costs, charges, or expenses; to include, charges made for travel and lodging.**

After the registration, we will send all qualified papers to the publish house and index organization for publishing directly.

We are looking forward to meeting all the authors in our conference.

Please strictly adhere to the format specified in the conference template while preparing your final paper. If you have any problem, please feel free to contact us via icprml@163.com. For the most updated information on the conference, please check the conference website at **http://www.prml.org/.** The Conference Program will be available at the website in middle **July, 2023.**

Again, congratulations. We are looking forward to seeing you in Urumqi, China..

Yours sincerely,

PRML 2023 Organizing Committees

http://www.prml.org/

PRML

UniQNatyam: An Approach Towards Non-Repetitive Dance Generation

Vibha Murthy
Computer Science and Engineering Department
PES University
Bengaluru, India
vibha.harsha@gmail.com

Vidisha Chandra
Computer Science and Engineering Department
PES University
Bengaluru, India
vidishasateesh@gmail.com

Vishakha Hegde
Computer Science and Engineering Department
PES University
Bengaluru, India
vishakhahegde23@gmail.com

Rayudu Srishti
Computer Science and Engineering Department
PES University
Bengaluru, India
rayudu.srishti7@gmail.com

K S Srinivas
Computer Science and Engineering Department
PES University
Bengaluru, India
srinivasks@pes.edu

***Abstract*— Dance is an art form involving body movements, expressions, and gestures to communicate emotions non-verbally. Choreographing a dance that aligns with the music is a complex process requiring time, money, and effort. This research paper aims to simplify choreography by exploring the relationship between music and dance motion. We propose using audio analysis, a cross-modal transformer, a reward model, and a 3D animated figure to generate unique and visually pleasing dances that accurately represent the music's characteristics. The goal is to create an immersive experience without repetitive movements while conveying the intended emotions and story.**

Keywords—Full Attention Cross-Modal Transformer, Reinforcement Learning, GAE, PPO, Advantage estimation.

I. INTRODUCTION

Traditionally, choreographers have relied on their creative intuition and expertise to design dance routines, a process that requires significant time, resources, and a deep understanding of both music and movement. However, these conventional methods often result in limitations such as repetitive patterns, lack of innovation, and difficulty in synchronizing complex movements with intricate musical arrangements. Choreographers are tasked with harmonizing movements with music, ensuring coherence, avoiding repetitiveness, and aligning dance steps with rhythm and emotion.

Our project aims to address the above issues by automating the choreography process by developing an automated choreographer that analyses songs, generates dance routines, and visualizes them on a 3D figure. By integrating the Full Attention Cross-Modal Transformer (FACT) model [1] with seed motions and a reward model, we ensure unique and non-repetitive choreography. This innovation has the potential to revolutionize the dance industry, saving time and resources while making dance accessible to a wider audience. With the COVID-19 pandemic disrupting live performances, our automated choreographer offers a solution for dancers to continue creating and producing remotely, helping to alleviate financial distress in the dance industry.

The global dance industry is estimated to be worth $10 billion, with projected growth to $14 billion by 2022. Despite a temporary setback in 2020 due to the pandemic, the market size of the dance studio sector in the United States is expected to rebound and reach $3.7 billion in 2022 [9]. By streamlining and automating the choreography process, our project has the potential to impact the dance industry by providing a cost-effective and accessible solution for creating engaging dance routines.

Our project utilizes the FACT model, which combines audio and motion transformers to encode the relationship between music and dance. By incorporating seed motions, the model ensures that the generated choreography aligns with the audio and maintains coherence throughout the performance. To avoid repetitive sequences, we employ Reinforcement Learning [8] through a reward model that predicts the likelihood of repetition in motion sequences. This adds depth and originality to the choreography, enhancing the viewer's experience.

The proposed automated choreographer benefits dancers of all levels, from beginners to professionals, by providing a tool for creating original and engaging dance routines. Additionally, it can be used in the context of musicals and theatre productions, where time and resource constraints often limit the creation of unique choreography. The system can also be valuable in educational settings, assisting in the training and refinement of dance skills for students. Overall, our automated choreographer has versatile applications in entertainment, education, and training within the dance industry.

To delve further into the details of our work, Section II of this paper explores the relevant literature in-depth, Section III discusses the dataset, Section IV presents the workflow and models used, Section V provides an extensive examination of our results and Section VI provides a user study of the results.

Furthermore, we conclude with a comprehensive summary of our findings and their implications for the future of dance choreography in Section VII.

II. Literature Survey

A. FACT

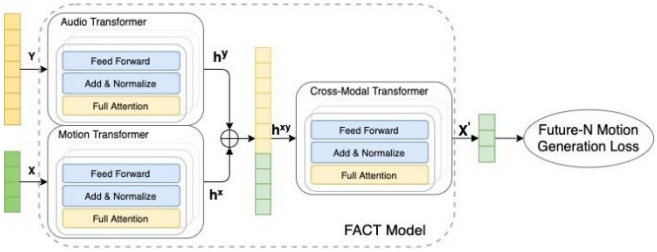

Fig. 1: Schematic diagram of FACT Model

Summary: Dancing is a universal language found in all cultures, and generating realistic 3D dance motions from music is a challenging task. In this work, a Full Attention Cross-modal Transformer (FACT) [1] network was proposed to address these challenges. The model generated a long sequence of realistic 3D dance motions from a short seed motion and audio features. FACT involved three key design choices, including full-attention mask, N future motion prediction, and cross-modal transformer. The motion quality, generation diversity, and motion-music correlation of the generated 3D dance motion were evaluated using Frechet Inception Distance (FID), average Euclidean distance, and Beat Alignment Score, respectively.

Methodology: The proposed FACT model encoded the seed motion and audio features using a motion transformer and audio transformer into motion and audio embeddings, respectively. These embeddings were concatenated and sent to a cross-modal transformer that generates N future motion sequences. The model was trained in a self-supervised manner and applied in an auto-regressive framework at test time. FACT used full-attention mask and predicts N future motions beyond the current input to pay more attention to the temporal context. Fig. 1. shows the structure of the 3 transformers including the cross modal transformer where in the audio and motion segments are concatenated and passed to it.

Evaluation: The generated 3D dance motion was evaluated using FID, average Euclidean distance, and Beat Alignment Score. The motion quality, generation diversity, and motion-music correlation of the generated motion were compared with three baselines, including GPT style causal transformer, motion encoder-decoder, and PCA-based method. The evaluation results show that the proposed FACT model generated 3D dance motion with high motion quality, generation diversity, and motion-music correlation.

Shortcomings: The proposed FACT model does not reason about physical interactions between the dancer and the floor, leading to artifacts such as foot sliding and floating. Also, the model is currently deterministic, and generating multiple realistic dance motions per music is an exciting direction for future research.

B. Bailando

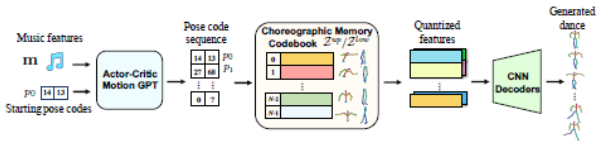

Fig. 2: Flow diagram of Bailando Model

Summary: Bailando [2] presented a novel approach for generating high-quality 3D dance sequences, meticulously synchronized with diverse music genres while adhering to stringent choreographic standards. To overcome the intricate challenges inherent in dance sequence generation, Bailando harnessed the capabilities of a choreographic memory and an Actor-critic Generative Pre-trained Transformer (GPT). This approach exhibited superior performance when compared to prior methods, showcasing promise for real-world applications. A reinforcement learning scheme was ingeniously integrated into the GPT model through the introduction of a specially devised beat-align reward function.

Methodology: Bailando's architecture was structured around two core components. The initial component involved the creation of a choreographic memory, accomplished using VQ-VAE to efficiently encapsulate reusable elements of dancing movements. This memory not only encapsulated the upper and lower compositional halves of 3D poses but also facilitated their quantization. The second pivotal element was the actor-critic GPT, adept at translating music and source pose codes into future pose codes. This GPT model was skillfully optimized through the Adam optimizer and trained with a multifaceted approach that included cross-conditional causal attention and the incorporation of a CNN decoder during the final stages of training.

Evaluation: Bailando's performance assessment yielded its most favorable results in terms of automatic metrics and visual judgments. Quality, quantified through Frechet Inception Distances (FID), exhibited a level of excellence, while diversity, determined by calculating the average feature distance of generated movements, confirmed the richness and variety of the output. Importantly, the generated dance sequences managed to encapsulate high-quality visual expressiveness and emotional resonance. Moreover, the learned choreographic memory effectively discerned human-interpretable dancing-style poses, all accomplished without the need for explicit supervision.

Shortcomings: Bailando's computational demands were significant, and the process of assembling authentic dancing clips as dance units demanded substantial manual effort. Moreover, the system displayed the ability to generate nonstandard poses that occasionally fell beyond the confines of the established dancing subspace, potentially resulting in performance instability. Additionally, the framework's current scope was restricted to generating 3D dance sequences, neglecting other essential elements of dance, such as facial expressions or clothing. As such, the avenue for further improvement remains open, particularly in terms of enhancing

the diversity and creativity of the generated dance sequences.

III. DATASET

We are using the AIST++ dataset to train the seed motion classifier. It is the largest known dance dataset and consists of 1408 dance sequences along with music. Fig. 3 depicts the structure of the AIST++ dataset including detailed description of dancers, choreographies, cameras and music involved.
We have 60 music clips ready that are classified into 10 music/ dance genres. The motion clips are between 7 seconds and 48 seconds. The 60 audios are different in terms

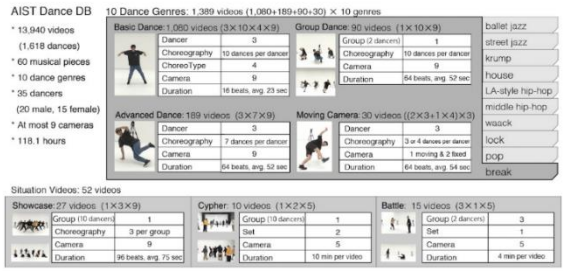

Fig. 3: AIST++ Dataset

of tempo and genres, and have multiple motion sequences relevant to a particular music clip.

For our implementation we have decided to use the SMPL format of motion sequences to represent the dance seed motions. The format consists of the following parameters:

Poses - (N,24,3) are the pose parameters

Trans - (N,3) is the 3D motion trajectory

Scaling - (N,1) is for the body scaling factor

The dataset has been split into train, validation and test sets.

IV. WORKFLOW AND MODELS USED

With the primary goal of reducing repetition in generated dance movements, we propose a reinforcement learning with human feedback approach to train the Full Attention Cross-Modal Transformer to guide it in generating dance sequences with less repetitive steps in a long-term order.
Reinforcement learning is an approach to train agents to take the right decisions based upon an external environment. The model receives rewards or penalties for its action and eventually fine-tunes its actions in order to maximize the rewards. This method has been proven to be effective in training models that perform a wide range of tasks from image generation to text generation to music piece generation.

We also propose a reward model as part of the environment that predicts the likelihood of repetition in the generated dance sequence. Utilizing the rewards or penalties that are given to the model, it performs backpropagation and optimizes the training process.

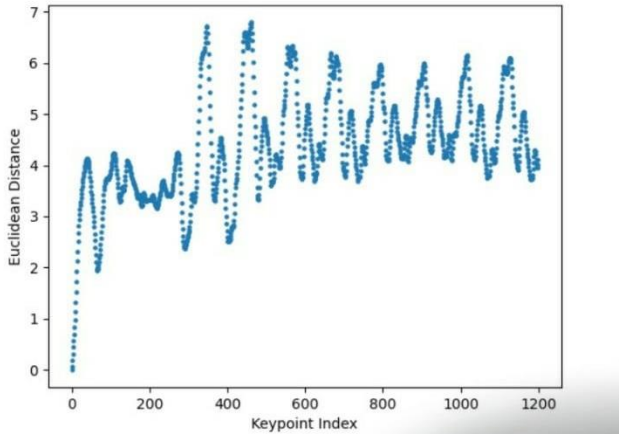

Fig. 4: Graph of Euclidean Distance vs Key point Index for a sample generated dance of the original FACT model (shows repetition of steps)

Reward Model with Human Feedback

We first train the reward model with a custom dataset that was created by sampling, modifying the AIST++ dataset in order to create two sets of motion sequences labelled with Repetitive or Non-Repetitive. This dataset consists of the same 10 genres present in the original dataset. The different choreographies are mixed and matched to create cohesive sequences of non-repetitive dance steps, and the original sequences are retained as repetitive. The non-repetitive sequences consist of one complete dance step (averaging around 120 frames or 2 seconds) followed by another complete dance step. The repetitive sequences have the same dance step that repeats twice or three times. Along with the above dataset, additional data is also generated using human feedback in the form of labels assigned to the generated motion sequences of the FACT model. Generated sequences that show relatively fewer repetitive steps are labelled non-repetitive, whereas steps that are mostly repeated are labelled repetitive. This process is performed for 500 generated sequences, with 350 being labelled repetitive and 150 being labelled non-repetitive. The skew in the dataset is balanced by the first method of synthesizing data from the AIST++ dataset.

To define what a repetitive sequence is, it comprises a set of steps that are repeated continuously and does not change as the music progresses. The graph in Fig 4. shows a sample of a generated dance where the Euclidean distance between each SMPL keypoint per frame of generated motion and the first one is plotted against the index of the keypoint. We can clearly see the pattern in steps repeating here. The aim was to generate non repetitive sets of dance conditioned on the same music denoted by Y = {y_1, y_2,…y_T'} change the

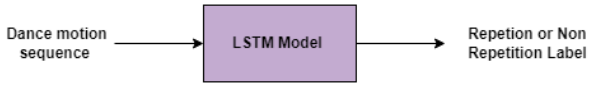

Fig. 5: LSTM Model Flowchart

dance step after one complete set is generated. The generated dance steps after the seed motion (denoted by Z = {z_1, z_2,…z_T}) are from T+1 to T' steps and denoted by A = {a_1, a_2,…a_T''}, B = {b_1, b_2,…b_T'} (2 sets of different dances generated in this case). This can be extended as the dance is generated in an autoregressive manner. The key concept is that the reinforcement learning constantly forces the dance generation to backtrack from the current state of the model and change the course of action when repetition is identified. Backtracking is possible as longer future motion sequences are predicted instead of short N motion sequences. During testing, the model generates the required dance steps including the change from one step to another.

Utilizing the above described dataset, we train an LSTM-based RNN with 128 hidden units. We denote the input motion sequence as:

$$X = \{x\_1, x\_2, \ldots, x\_T\}, \quad (1)$$

where x_t represents the SMPL keypoints at time step t.

Each frame of the motion sequence is input sequentially into the model. Each LSTM cell consists of a hidden state h_t and a cell state c_t. At each time step t, the LSTM updates the hidden state and cell state based on the input at that
time step and the previous hidden state and cell state:

$$h\_t, c\_t = LSTM(x\_t, h\_\{t-1\}, c\_\{t-1\}) \quad (2)$$

The output of the LSTM at the last time step T is denoted as h_T. The output of the last LSTM cell is sent to a fully connected layer with weights W and bias b:

$$z = W * h\_T + b \quad (3)$$

The sigmoid activation function is applied to obtain the predicted output:

$$y\_pred = sigmoid(z) \quad (4)$$

The loss is computed using binary cross-entropy, comparing the predicted output to the true labels:

$$loss = -[y\_true * log(y\_pred) + (1 - y\_true) * log(1 - y\_pred)] \quad (5)$$

Backpropagation is performed using Adam optimization algorithm. The train + validation splits are utilized. For evaluation we use the test split and feed each pose sequence into the LSTM and obtain the predicted output. Fig. 5 depicts the workflow for the LSTM model. We then compare the predicted output to the true labels to evaluate the model's performance. The model performed with an accuracy of 79%.

The negative log likelihood of repetition is taken from this model to use as a reward to the dance generation model. In effect, this negative log likelihood must be maximized in order to generate dance that is less repetitive.

$$reward = -log(predicted\_repetition\_likelihood) \quad (6)$$

Workflow

In order to train the dance generation model, we perform reward prediction for motion sequences of N steps length as one time step. The Advantage estimate of the generated motion sequences is then calculated for the specific audio, using the value function (expected sum of future negative log likelihood of predicted repetition), following which, a PPO mechanism is deployed to perform updates over mini batches of audio, motion sequences and advantage estimates. Refer to Fig. 6 for visualization of the workflow.

Generalized Advantage estimation (GAE)

Input: rewards, values, γ
T = length of rewards (total number of time steps)
advantages = array of zeros with length T
Δ = rewards + γ*append(values[1:], values[-1]) - values
accumulated_advantage = 0
for t in reversed(range(T)):
    accumulated_advantage = Δ[t] +
    γ*accumulated_advantage
    advantages[t] = accumulated_advantage
Calculate μ and σ of advantages:
total_sum = 0
for t in range(T):
    total_sum = total_sum + advantages[t]
μ = total_sum / T
$\sigma^2 = 0$
for t in range(T):
    $\sigma^2 = \sigma^2$+ (advantages[t] - μ) * (advantages[t] - μ)
$\sigma = \sqrt{\sigma 2 / T}$
Normalize advantages:
for t in range(T):
    advantages[t] = (advantages[t] - μ) / σ
Output: advantages

We compute the rewards R(t) for each time step (N frames of motion sequence) using our reward model. We compute the value function V(t) for each time step which estimates the expected sum of future rewards starting from time step t.

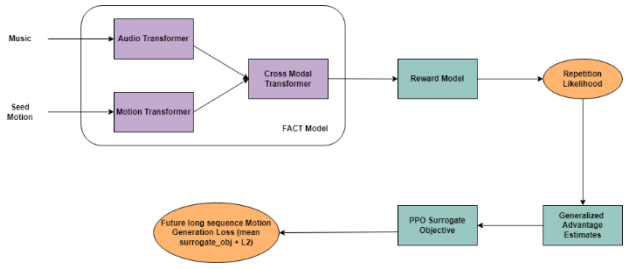

Fig. 6: Flowchart of FACT Model along with Reward Mechanism, PPO and GAE

We then compute the advantage estimates A(t) for each time step. The advantage represents how much better or worse a subsequence is compared to the average generated sub sequences.

$$A(t) = \sum[i = t\ to\ T-1]\ \gamma^{(i-t)} * \delta(i) \quad (7)$$

where T is the total number of time steps, γ is the discount factor ($0 \leq \gamma \leq 1$), and δ(i) is the temporal difference error defined as:

$$\delta(i) = R(i) + \gamma * V(i+1) - V(i) \quad (8)$$

The temporal difference error δ(i) captures the difference between the expected value of the current sub sequence and the value of the next sub sequence.

We then accumulate the advantages for each subsequence and normalize them by subtracting the mean and dividing by the standard deviation of the advantages across all time steps to improve stability.

PPO

Proximal Policy Optimization, is an algorithm designed to train policies for agents in environments where actions are taken sequentially to maximize rewards. In the context of dance generation, the model must generate dance that in the long run is less repetitive. Hence after each dance step, the model must decide to change the course of action and generate different frames that amount to a different dance step while still being able to coordinate the kinematic beats with the music beats. The FACT model's autoregressive nature and structure handles the music to motion mapping and the reinforcement learning approach ensures less likely repetition of steps in a longer dance sequence generation.

```
for epoch in range(num_epochs):
    for batch_audio, batch_motion, batch_advantage in dataset:
        log_probs=model.compute_log_probs(batch_motion)
        ratio = exp(log_probs - batch_advantage)
        surrogate_obj=min(ratio*batch_advantage, clamp(ratio,
        1 - ε, 1 + ε) * batch_advantage)
loss = -mean(surrogate_obj)
```

We compute the log probabilities to evaluate the likelihood of the generated motion sequences given the audio input. By calculating the log probabilities, the model can quantify how probable or likely each generated motion sequence is under its current policy.

We then calculate the ratio of the log probabilities of the current policy to the old policy. We finally compute the surrogate objective for PPO by taking the minimum of the ratio multiplied by the advantage and a clipped version of the ratio multiplied by the advantage. The surrogate objective function places a constraint on the change between the new and old policies, and prevents large policy updates that might lead to instability.

The loss function for the model is defined as the mean of the surrogate objective of each batch in the entire dataset. This loss is taken into consideration along with the L2 loss of the full attention cross modal transformer network.

We perform backpropagation in a self-supervised manner using the combined loss value.

V. RESULTS

We utilize two main evaluation functions to determine the quality of dance generation and amount of repetition.

To evaluate the goodness of predicted dance moves using the FACT model with a reward function, we considered using the Fréchet Inception Distance (FID) score as a loss function.

We denote the feature representations of the real generated moves as μ_r and Σ_r, representing the mean and covariance matrices, respectively. Similarly, let μ_g and Σ_g represent the feature statistics of the generated moves. The FID score is then calculated as:

$$FID = ||\mu\_r - \mu\_g||^2 + Tr(\Sigma\_r + \Sigma\_g - 2(\Sigma\_r\Sigma\_g)^{1/2}) \quad (9)$$

TABLE I: Comparison of evaluation metrics

Solution	Motion Quality FIDk	Motion Quality FIDg	Motion Diversity Dist.	Motion-Music Corr Beat Align	User Study Avg Rating [1-5]
Li et al. [7]	86.43	6.85	6.85	.232	2.5
Dance net [6]	69.18	2.86	2.86	.232	3.25
Dance Revolution [5]	73.42	3.52	3.52	.220	3
FACT [1]	35.35	5.94	5.94	.241	3.5
UniQNatyam (ours)	**31.8**	**6.23**	**6.36**	**.243**	**4.25**

As seen in Table I, The FID (kinetic) value for the FACT model alone was about 35.35 whereas our model produces a lower value of 31.8. The lower the FID score, the better the generated movements align with the real style. Motion diversity also seems to be improved (6.36 over 5.94 of FACT model) due to non-repetition of steps as the measure takes an average of deviation in motions in a length of a sequence. Our model proves to have less repetition and more dynamic movements which improves the choreography to a great extent due to the reward function performed.

The likelihood of repetition of the entire generated sequence is retrieved using the previously stated reward model. 500 sequences from across different genres were taken for testing. We obtained an average of 77% non-repeated sequences after performing tests.

VI. User Studies

We conducted a user study to assess the fluidity and non-repetition of the dance sequences generated by our model by comparing it with other existing models. Over 35 participants were asked to rate each dance sequence on a scale of 1 to 5. Table II depicts the results of the study. Our dance sequences were commended for their avoidance of repeated steps, indicating the diversity and richness of the choreography produced by our model.

TABLE II: Result of the user studies done on FACT and UniQNatyam

Solution	Fluidity	Non - Repetition
FACT [1]	3.8	2.5
UniQNatyam(ours)	3.9	3.2

VII. Conclusion

By incorporating Reinforcement Learning along with the excellent FACT model in our project, the dance that is generated is of high quality. The dance that the model autoregressively generates does an apt music to motion mapping, making sure each step that is generated sits well with the beat of the provided song. Our Reinforcement Learning approach ensures that the steps that the model generates are not monotonous and repetitive. Applications are numerous and can be used in small-scale movie industries as well as dance classes in order to weave stories in the form of dance. We further aim to implement aspects of direction, background and lighting for dance performances generated to provide a more wholesome production.

REFERENCES

[1] Li R, Yang S, Ross DA, Kanazawa A. “AI Choreographer: Music Conditioned 3D Dance Generation with AIST++”. vis IEEE/CVF International Conference on Computer Vision (ICCV) 2021, pp. 13401-13412.

[2] Li Siyao, Weijiang Yu, Tianpei Gu, Chunze Lin, Quan Wang, Chen Qian, Chen Change Loy, Ziwei Liu, “Bailando: 3D Dance Generation by Actor-Critic GPT with Choreographic Memory” vis ICCV 2022, pp. 11050-11059.

[3] Ruozi Huang, Huang Hu, Wei Wu, Kei Sawada, Mi Zhang, Daxin Jiang., “Dance revolution: Long-term dance generation with music via curriculum learning” vis International Conference on Learning Representations, 2020.

[4] Wenlin Zhuang, Congyi Wang, Siyu Xia, Jinxiang Chai, Yangang Wang. “Music2dance: Music-driven dance generation using wavenet” vis ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM) 2022.

[5] S. Geetha, G. Poonthalir, and P. T. Vanathi, "A Hybrid Particle Swarm Optimization with Genetic Operators for Vehicle Routing Problem," Journal of Advances in Information Technology, Vol. 1, No. 4, pp. 181-188, November, 2010.doi:10.4304/jait.1.4.181-188

[6] Hüseyin Demirci, Ahmet Turan Özcerit, Hüseyin Ekiz, and Akif Kutlu, "Knowledge-Based System Framework for Training Long Jump Athletes Using Action Recognition," Vol. 6, No. 4, pp. 217-220, November, 2015. doi: 10.12720/jait.6.4.217-220

[7] Jenn-Long Liu, Chung-Chih Li, and Chien-Liang Chen, "Local Search-based Enhanced Multi-objective Genetic Algorithm and Its Application to the Gestational Diabetes Diagnosis," Vol. 6, No. 4, pp. 252-257, November, 2015. doi: 10.12720/jait.6.4.252-257

[8] A. Hammoudeh, “A Concise Introduction to Reinforcement Learning”, Princess Suamaya University for Technology: Amman, Jordan, 2018.

[9] Statista Research Department. "Market size of the dance studio sector in the United States from 2012 to 2021, with a forecast for 2022." https://www.statista.com/statistics/1175824/dance-studio-industry-market-size-us/ (published Oct 28, 2022)