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*

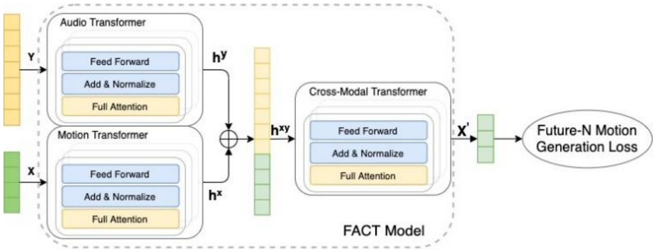


Fig. 1: Schematic diagram of FACT Model

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*

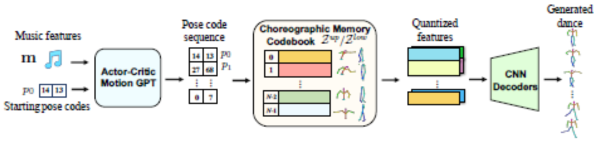


Fig. 2: Flow diagram of Bailando Model

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.

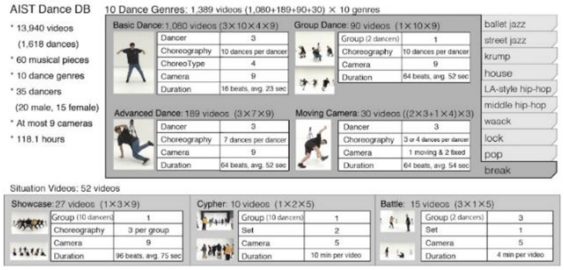
Methodology: Bailando's architecture was structured around two core components. The initial component involved the creation of a choreographic memory, accomplished using VQ-VAE to efficiently encapsulate reusable elements of dancing movements. This memory not only encapsulated the upper and lower compositional 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 of tempo and genres, and have multiple motion sequences relevant to a particular music clip.



AIST Dance DB

- 13,940 videos (1,618 dances)
- 60 musical pieces
- 10 dance genres
- 35 dancers (20 male, 15 female)
- At most 9 cameras
- 118.1 hours

10 Dance Genres: 1,389 videos (1,080+189+90+30) × 10 genres

Basic Dance: 1,080 videos (3×10×4×9)

Dancer	3
Choreography	10 dances per dancer
ChoreoType	4
Camera	9
Duration	16 beats, avg. 23 sec

Group Dance: 90 videos (1×10×9)

Group (2 dancers)	1
Choreography	10 dances per dancer
Camera	9
Duration	64 beats, avg. 52 sec

Advanced Dance: 189 videos (3×7×9)

Dancer	3
Choreography	7 dances per dancer
Camera	9
Duration	64 beats, avg. 52 sec

Moving Camera: 30 videos ((2×3+1×4)×3)

Dancer	3
Choreography	3 or 4 dances per dancer
Camera	1 moving & 2 fixed
Duration	64 beats, avg. 54 sec

Situation Videos: 52 videos

Showcase: 27 videos (1×3×9)

Group (10 dancers)	1
Choreography	3 per group
Camera	9
Duration	96 beats, avg. 75 sec

Cypher: 10 videos (1×2×5)

Group (10 dancers)	1
Set	2
Camera	5
Duration	10 min per video

Battle: 15 videos (3×1×5)

Group (2 dancers)	3
Set	1
Camera	5
Duration	4 min per video

Fig. 3: AIST++ Dataset

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.

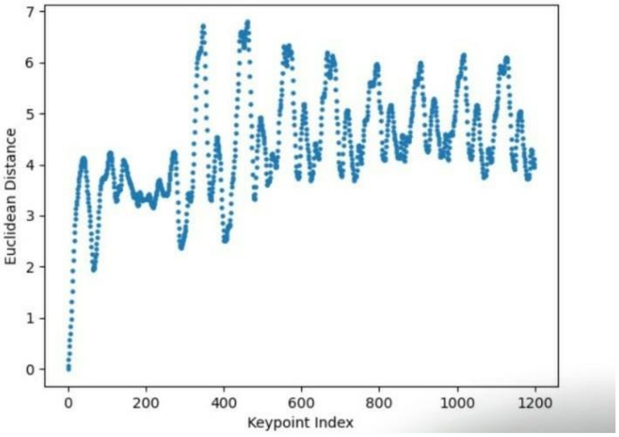


Fig. 4: Graph of Euclidean Distance vs Key point Index for a sample generated dance of the original FACT model (shows repetition of steps)

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

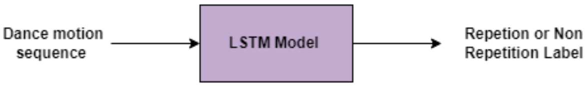


Fig. 5: LSTM Model Flowchart

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:

$$X = \{x\_1, x\_2, \dots, x\_T\}, \tag{1}$$

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:

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

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:

$$z = W * h\_T + b \tag{3}$$

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

$$y\_pred = sigmoid(z) \tag{4}$$

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

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

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.

$$reward = -log(predicted\_repetition\_likelihood) \tag{6}$$

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.

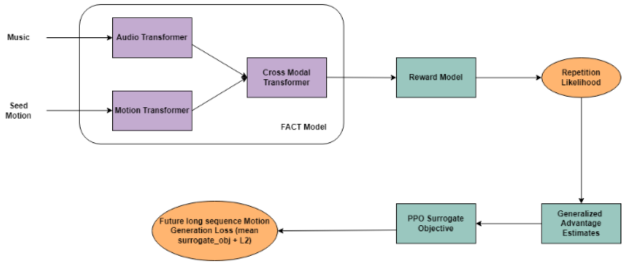


Fig. 6: Flowchart of FACT Model along with Reward Mechanism, PPO and GAE

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.

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

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:

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

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:

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

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)