DanceCraft: A Music-Reactive Real-time Dance Improv System

Ruilin Xu
rxu@cs.columbia.edu
Columbia University
New York, NY, USA

Vu An Tran
tranvuan82@gmail.com
Snap Inc.
New York, NY, USA

Shree K. Nayar
nayar@cs.columbia.edu
Columbia University
New York, NY, USA

Gurunandan Krishnan
guru@snap.com
Snap Inc.
Bellevue, WA, USA

ABSTRACT

Automatic generation of 3D dance motion, in response to live music, is a challenging task. Prior research has assumed that either the entire music track, or a significant chunk of music track, is available prior to dance generation. In this paper, we present a novel production-ready system that can generate highly realistic dances in reaction to live music. Since predicting future music, or dance choreographed to future music, is a hard problem, we trade-off perfect choreography for spontaneous dance-motion improvisation. Given a small slice of the most recently received audio, we first determine where the audio include music, and if so extract high-level descriptors of the music such as tempo and energy. Based on these descriptors, we generate the dance motion. The generated dance is a combination of previously captured dance sequences as well as randomly triggered generative transitions between different dance sequences. Due to these randomized transitions, two generated dances, even for the same music, tend to appear very different. Furthermore, our system offers a high level of interactivity and personalization, allowing users to import their personal 3D avatars and have them dance to any music played in the environment. User studies show that our system provides an engaging and immersive experience that is appreciated by users.

CCS CONCEPTS

• **Computing methodologies** → **Animation**; **Neural networks**.

KEYWORDS

music-reactive, real-time, 3D dance generation, motion in-betweening, deep learning

ACM Reference Format:
Ruilin Xu, Vu An Tran, Shree K. Nayar, and Gurunandan Krishnan. 2024. DanceCraft: A Music-Reactive Real-time Dance Improv System. In *9th International Conference on Movement and Computing (MOCO '24), May 30–June 02, 2024, Utrecht, Netherlands.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3658852.3659078

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
MOCO '24, May 30–June 02, 2024, Utrecht, Netherlands
© 2024 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 979-8-4007-0994-4/24/05
https://doi.org/10.1145/3658852.3659078

1 INTRODUCTION

Virtual Reality (VR) and Augmented Reality (AR) technologies are rapidly gaining prominence in transforming real world entertainment into immersive digital experiences. This is particularly evident in the realm of virtual concerts and dance performances. Platforms such as Fortnite, Snapchat, and TikTok are leading this digital revolution. Users, using their virtual avatars as proxies, can interact with live music. Animating their avatars with realistic dance movements in response to music enhances the engagement.

Traditionally, dance animation production has been a manual and labor-intensive process. Choreographers and dancers work in tandem to create dance routines that resonate with various music genres, tempos, and styles. These performances are then captured and digitized by specialized teams. Although this method can yield high-quality results, it is both resource-intensive and time-consuming, highlighting the need for more efficient solutions in the fast-paced digital entertainment landscape.

Recent advancements in deep learning have enabled the synthesis of music-conditioned dances, as demonstrated in [8, 31, 47, 53, 82, 84]. These methods replicate the traditional choreography process, aiming to align dance movements with musical rhythms and styles [19, 31, 35, 91]. However, these approaches often lack realism and tend to overlook crucial human elements of dance, such as variety, expressiveness, spontaneity, and personal style, thus diminishing user engagement. We aim to address these shortcomings, provide an immersive and realistic dance experience and foster an intimate connection between users and their dancing avatars. This leads us to a pivotal question: *What forms of dance, beyond strictly choreographed routines, can enhance the personalization and interactivity of user experiences?*

Improvisational dance, known for its spontaneous and unstructured nature, offers a wide range of unique interpretations to the same piece of music, thereby enriching the digital user experience [58, 67, 75]. This form of dance, deeply embedded in everyday practices, not only promotes artistic expression but also strengthens the connection between performer and audience [63]. In contrast, choreographed dance involves a precise and methodical process, where dancers strictly adhere to predefined movements, limiting spontaneous expression [14].

This paper introduces a novel system designed for synthesizing improvisational dance in real-time to live music, with the following key contributions and advantages:

(1) Hybrid Modeling Approach: We introduce a hybrid model that merges data-driven graph-based methods with deep learning techniques for creating music-conditioned 3D improvisational dance. This process begins with the extraction of various music descriptors in response to a music signal. These descriptors guide the selection of dance segments from a database, from which we randomly choose one to match the music's characteristics.

The selected dance segment is then fused into the ongoing animation using a lightweight state-of-the-art motion in-betweening network. This approach ensures fluid transitions between dance routines, allowing for a seamless and dynamic flow of movements that adapt in real-time to the music.

The resulting dances are not only spontaneous and engaging but also maintain a strong coherence with the music's genre, tempo, and energy. This balance of unpredictability and musical alignment reflects the essence of improvisational dance.

(2) Comprehensive 3D Dance Dataset: To tackle the scarcity of diverse dance data, we've compiled an extensive dataset of high-quality dance animations. This collection spans over 8 hours and encompasses a broad spectrum of musical tempos, energies, and genres. Furthermore, our dataset is enriched with natural idle animations and a variety of facial expressions, significantly boosting the avatars' expressiveness and emotional connection during their dance performances.

(3) Interactivity and Customization: The music that animates the dances can be sourced from various inputs, such as the end user or a DJ. When users select their own music, they transform from passive viewers into active participants, having direct control over the music and, consequently, the dance animations. For DJs or artists, our system offers the ability to associate popular and recognizable choreographed dance routines with specific music segments, thereby heightening the user's personal engagement and experience.

(4) Personal Avatar Integration: Our system stands out by allowing users to import their personal 3D avatars from popular AR/VR platforms, instead of limiting them to a few predefined options. We currently support Bitmojis [1], a 3D avatar used by over 250 million Snapchat users [2]. We believe this fosters a deeper connection between users and their dancing avatars.

(5) Integrated Production-Ready System: We detail both the hardware setup and software implementation of our improvisational dance system. We personalize the user experience by allowing users to cast their Bitmoji 3D avatars by scanning a QR code. Our system is versatile – designed as a web service, it is suitable for deployment on individual kiosks or for large online virtual gatherings. We will release the code and datasets in the future.

Empirical evidence from extensive user studies confirm the effectiveness of our system in delivering an enjoyable, engaging, and highly personalized virtual dance experience.

2 RELATED WORK

Human Motion Synthesis. Human motion synthesis aims to generate natural human movements based on existing data, accounting for the non-linear and stochastic nature of human motion. Initial approaches utilized classical techniques like hidden Markov models [11, 12, 52, 83] and statistical models [15, 46, 66]. Recent advancements have seen the application of neural networks, training on 3D human motion datasets to generate motion using various architectures, including CNNs [23, 29, 30], GANs [73], RNNs [7, 8, 13, 20, 21, 34, 44, 60, 65, 89], and transformers [6, 18, 24, 26, 38, 55].

Graph-Based Motion Synthesis. Graph-based methods in human motion synthesis involve constructing motion graphs, where nodes represent motion segments, and edges indicate transition probabilities. Pioneered by Lamouret et al. [45], this approach has evolved with contributions from [9, 42, 43, 48, 50, 61, 74]. Recent extensions [40, 41, 77] integrate rhythmic constraints, while others [10, 35, 56] incorporate complex choreographic rules.

Motion In-betweening. Initially, motion in-betweening methods employed linear or spline interpolation techniques to generate intermediate frames between keyframes [62, 72]. Advanced deep learning methods, especially RNNs, now dominate this space. Harvey et al. [25] introduced the Recurrent Transition Network (RTN) with subsequent enhancements [26]. Other approaches involve RNN and CVAE combinations [85], convolutional autoencoders [37, 73, 95], transformers [18, 26, 38, 64, 69], and diffusion models [28, 39, 70, 80, 86, 87, 94].

Dance Motion Synthesis with Music. Dance motion synthesis, a subset of human motion synthesis, initially focused on similarity-based matching [51, 77]. Crnkovic-Friis et al.[17] introduced LSTM-based deep learning methods, followed by various other LSTM applications [8, 36, 84, 90]. GANs [47, 54, 82] and transformers [31, 53, 55, 78, 79] represent other explored architectures. GrooveNet [8] is noteworthy for real-time music-driven dance synthesis, although its scope and user engagement are limited.

3D Dance Dataset. Existing 3D motion datasets focus mainly on daily human motions [16, 32, 33, 59, 76]. Dance-specific datasets, often created from 2D-to-3D conversions, lack accuracy and often result in unnatural or even physically implausible motions [47, 49, 53, 55, 55, 82, 88]. Direct motion capture has been used to overcome these limitations [8, 84, 96], yet these efforts are limited in scope and diversity. Our research necessitates a comprehensive 3D dance dataset aligned with music tracks across various genres, necessitating the collection of our dataset.

3 EXPERIENCE DESIGN

In this section, we detail the foundational principles of our proposed dance experience. In envisioning a generative dance system, our goals are aligned with the principles of improvisational dance, while striving to create a highly engaging and interactive user experience. These goals include:

- **Enhanced User Interactivity:** The system should allow users to influence the dance animations by controlling the music, encouraging active engagement.
- **Real-Time Spontaneity:** The ideal system must react instantaneously to user interactions, requiring real-time music analysis and dynamic dance synthesis.
- **Element of Surprise:** Incorporating unpredictability and uniqueness in each dance sequence to reflect the essence of improvisation.

- **Physical Realism:** Generated dance motions should be realistic and fluid, free from artifacts typical to deep learning based systems.
- **Natural Avatar Behavior:** In the absence of music, avatars should exhibit natural, idle behaviors to enhance realism.
- **Ease of Extensibility:** The system should be easily updated with new dance routines without retraining.
- **Dance Authoring:** The system should allow artists to inject specific dance routines to particular musical segments. Ex: Macarena dance, Gangnam Style dance.
- **Avatar Personalization:** The system should support a user's personal 3D avatar from various platforms, deepening user connection and immersion in the virtual dance environment.
- **Animation Personalization:** Avatars come in various sizes and shapes. The synthesized dance should preserve the motion semantics between various avatars while minimizing artifacts.

4 METHOD

In this section, we present `Dance Synthesizer`, our proposed approach that takes user's music of selection as input and generates music-conditioned 3D improvisational dance.

4.1 Overview

Currently, a prevalent class of methods is to generate dance frames by sampling from a learned dance motion manifold conditioned on the entirety of the input music [8, 19, 31, 47, 82]. However, these methods heavily depend on the quality of learned manifold which requires a substantial amount of data. Even then, generated dances often lack of realism and tend to be repetitive. In addition, none of these methods can perform in real-time, reactive to user interactions. To overcome these limitations, our `Dance Synthesizer` is divided into two distinct components: music analysis and dance synthesis. Each of these components is discussed in details below.

4.2 Music Analysis

Upon user selection, the chosen music track is fed into the Music Analysis component for real-time analysis. We employ a moving window approach with a 3-second window size and 0.3-second increments. Within each window, the system analyzes the music across three key dimensions: energy, tempo, and musical type. These dimensions are explained in detail below.

Music energy: We define the music energy as the loudness of the music signal. According to Steven's power law [81], the loudness of a music signal is its intensity raised to the power of 0.67. Please note that this is a rough estimation of the music's energy, as it is computationally inexpensive and sufficient for our needs.

Tempo estimation. We utilize BeatNet [27], a state-of-the-art method, for real-time tempo estimation. Specifically, we apply BeatNet's "offline mode" on each 3-second music window, updated every 0.3 seconds. Finally, we employ a filtering process to minimize abrupt variations in the final tempo estimation results.

Music type estimation. We categorize music into three types: idling (absence of music), slow-paced, and fast-paced music, similar to [96]. To identify these three music types in real-time, we utilize YAMNet [68], a light-weight network known for its great performance on sound event classification. We train the network on a dataset created using Jingle Punks [3], with labels corresponding to the three defined categories: 0 for idling, 1 for slow-paced, and 2 for fast-paced music. Furthermore, we integrated YAMNet's original version on sound event classification to help distinguish between music, background noise, and human speech. With these two versions of YAMNet together, our system is able to accurately classify both music existence and music types in real-time.

4.3 Dance Synthesis

Subsequently, the Dance Synthesis component of our system generates dance movements in real-time, aligning with the input music's characteristics determined by the Music Analysis component. Unlike the limitations of existing methods discussed in Section 4.1, our approach instead takes inspiration from the classic graph-based methodologies.

4.3.1 Inspiration: classic graph-based approaches. This class of methods synthesize motion sequence by finding an optimal path within a pre-constructed motion graph. In such a graph, nodes represent individual motion segments from a database, and edges denote the probabilities of transitioning between these segments. The calculation of these probabilities could involve, for instance, assessing the similarity between ending motion segments of the two connected nodes.

However, a limitation of this approach lies in the potential sparsity of the motion graph. A sparse graph, with insufficient amount of edges connecting motion segments, can lead to repetitive dance sequences, contrary to our goal of generating diverse and spontaneous dances. Additionally, no music information can be incorporated in these methods. Therefore, to overcome these issues, our Dance Synthesis component employs a hybrid approach, combining the concept of a classic motion graph with a learning-based method.

4.3.2 Our hybrid approach. Aligning with the classic graph-based methods, our approach also synthesizes dance movements by connecting various dance motion segments. The key distinction lies in the creation of edges: unlike classic methods that establish edges between segments based on specific criteria, we connect segments based on the music information retrieved in real-time, as detailed in Section 4.2. For instance, when the user initiates a music track, our Dance Synthesis component immediately connects the current idle motion to any one of the dance segments that match the music's characteristics. This approach yields a highly dense motion graph, providing an almost limitless array of paths for dance synthesis.

Given this density, our method extends beyond the traditional confines of a motion graph. In this case, our hybrid approach essentially becomes a task in motion in-betweening, focusing on generating dance transitions between two motion segments that are expressive and diverse.

4.3.3 Dataset. Since our hybrid approach adapts a modified version of the motion graph, it is necessary to create a comprehensive and diverse dataset for the approach to run effectively. We engaged professional dancers to perform improvisational dances to an extensive selection of over 200 songs. These tracks were meticulously chosen from a variety of genres, including hip-hop, country, folk,

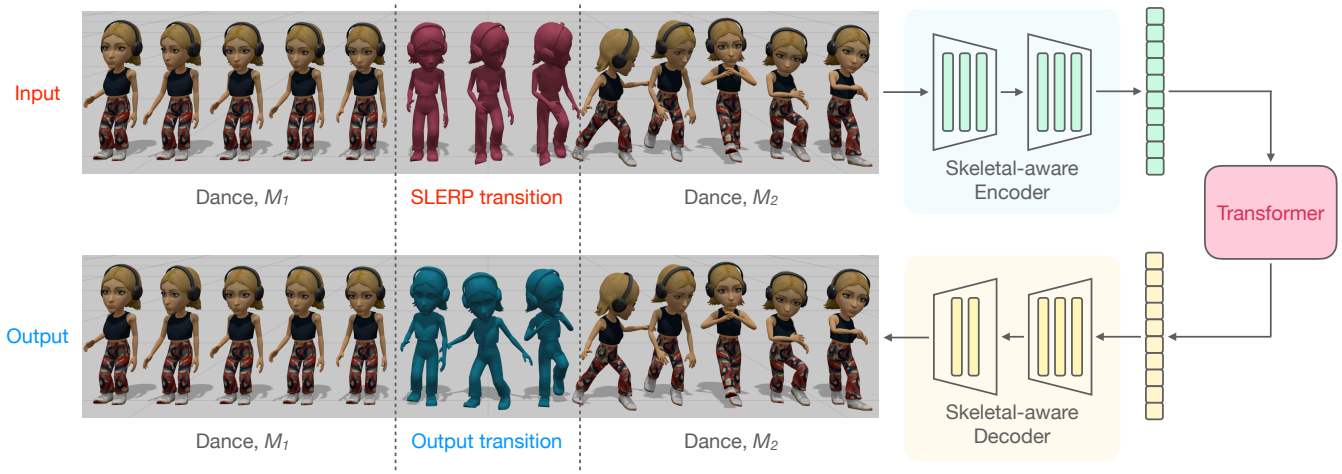


Figure 1: *Our dance transition generator.* The transition generator, $\mathcal{G}_T$, has three components: (a) a skeletal-aware encoder using skeletal operators; (b) a transformer subnetwork that operates in the latent space generated by the encoder; and (c) a skeletal-aware decoder that projects latent vectors back to 3D skeletons.

jazz, and disco, to ensure a broad representation of musical styles. The tempos of these songs varied from 80 BPM to 160 BPM, to accommodate a diverse set of dance rhythms and patterns.

In order to capture the dance movements with high fidelity, we equipped the dancers with the Smartsuit Pro II from Rokoko [71]. This motion capture suit allowed us to record the dances directly in 3D, ensuring the accuracy and quality of the captured motions. The result of this process is a comprehensive collection of over 6 hours of dance movements, stored in the Mixamo format [32].

A crucial aspect of our dataset is its versatility to accommodate a wide range of body shapes, from skinny to obese. Leveraging retargeting techniques inspired by [93], all captured dance movements to six distinct body shapes. This flexibility ensures seamless integration of different Bitmojis into our system.

4.3.4 Approach details. Here we introduce the implementation details of our hybrid approach, specifically, the transition generation network which we denote as $\mathcal{G}_T$.

$\mathcal{G}_T$ vs. typical motion in-betweening. Typical motion in-betweening methods such as [26, 38], concentrate on generating transitions from an initial motion sequence to a designated *target pose*. In this case, these methods only need to ensure the smoothness between the initial motion sequence to the transition, as long as the transition ends in the target pose. In contrast, $\mathcal{G}_T$ aims to ensure smooth motion continuity not only from the initial sequence to the transition movements but also from the transition to subsequent dance motions which could originate from a different dance track with potentially contrasting physical movements.

Network input. The input for the proposed $\mathcal{G}_T$ consists of a sequence of motion frames divided into three parts: (1) an initial dance segment from the source track, (2) placeholders for intermediate transition frames, and (3) a dance segment from the target track, which may differ from the source. We set the length of the input sequence as $L = M_1 + T + M_2 = 160$ frames, with $M_1 = M_2 = 65$ frames each for the two dance segments, and $T = 30$ frames for the transition, equivalent to one second of dance motion at the system's frame rate of 30 FPS. These 30 placeholder frames are initially generated using Linear Interpolation (LERP) for the global root position and Spherical Linear Interpolation (SLERP) for each joint's rotation between the last frame of the initial segment and the first frame of the subsequent one. $\mathcal{G}_T$ then learns to refine these placeholder frames, ensuring a seamless and natural connection between the dance segments on each end.

Network architecture. The network architecture of $\mathcal{G}_T$ draws inspiration from the framework of a denoising autoencoder. In this configuration, the network is trained to accurately predict a ground-truth motion sequence from an input sequence that has been altered or perturbed. As depicted in Figure 1, our network features a skeletal-aware encoder consisting of two skeletal blocks [5]. Each block is composed of a skeletal convolution, an activation function, and skeletal pooling, which collectively map each input motion frame into a corresponding latent vector. The skeletal-aware decoder, mirroring the encoder, also comprises two skeletal blocks. These blocks function to project the latent vectors back into the domain of 3D skeletal motions, thereby reconstructing the motion sequence.

A key distinction of our network model from the conventional denoising autoencoders is the integration of a transformer subnetwork positioned between the encoder and decoder. This transformer subnetwork, based on the structure from [38], has been adapted to accept inputs from our skeletal-aware encoder and produce outputs compatible with our skeletal-aware decoder. The inclusion of this transformer subnetwork is critical in our architectural design, as it is essential in generating an intermediary motion sequence that is both engaging and relevant to the ongoing dance synthesis. This is confirmed by the results of the ablation study discussed in Section 6.1.

4.3.5 Data and training. Here we discuss the data and representation used to train $\mathcal{G}_T$ and its training objectives.

Data. To train $\mathcal{G}_T$, we randomly selected approximately one-tenth of the total number of dance tracks from the comprehensive dataset introduced in Section 4.3.3.

Data representation. In our system, the 3D human skeleton motions is represented as two components: a static component and a dynamic component. The static component, denoted as $\mathbf{S} \in \mathbb{R}^{J \times S}$, encapsulates the armature information, where J represents the number of armatures and $S = 3$ corresponds to the 3D spatial coordinates. The dynamic component, symbolized as $\mathbf{Q} \in \mathbb{R}^{L \times J \times Q}$, captures motion dynamics, with L indicating the length of the motion sequence and $Q = 4$ reflecting the use of Quaternions for rotation representation. Additionally, the root joint is represented as $\mathbf{R} \in \mathbb{R}^{L \times (S+Q)}$, distinct from the J armatures. It comprises a sequence of global translations and rotations, crucial for maintaining the integrity and coordination of the entire skeletal motion.

This approach to data representation lays the groundwork for implementing *skeleton-aware* operations in $\mathcal{G}_T$. These operations, derived from the graph-based structural representation of the skeleton, consider key aspects such as bone hierarchy and joint adjacency. Such consideration is vital for ensuring the physical plausibility of generated motions. For a comprehensive understanding of these skeleton-aware operations and their implications in motion synthesis, we direct readers to the foundational work by Aberman et al. [5], which delves into the intricacies of skeletal convolution and skeletal pooling.

Training objectives. Let E, T, and D denote the encoder, transformer, and decoder components of $\mathcal{G}_T$, respectively. We optimize the following loss function:

$$\begin{aligned}\mathcal{L} = &\mathbb{E}_{(\mathbf{S},\mathbf{Q}_i)\sim\mathcal{M}}\left[\left\|\left(D(T(E(\mathbf{S},\hat{\mathbf{Q}}_i)),\mathbf{S}),\mathbf{S}\right) - \mathbf{Q}_i\right\|_2\right] \\ &+ \mathbb{E}_{(\mathbf{S},\mathbf{Q}_i)\sim\mathcal{M}}\left[\left\|\mathrm{FK}\left(D(T(E(\mathbf{S},\hat{\mathbf{Q}}_i)),\mathbf{S}),\mathbf{S}\right) - \mathbf{P}_i\right\|_2\right],\end{aligned} \tag{1}$$

where $\mathcal{L}$ is a standard reconstruction loss over the joint rotations and joint positions. Each frame of motion $i \in \mathcal{M}$ is represented by $(\mathbf{S}, \mathbf{Q}_i)$, the pair of skeleton offset and joint rotation. $\hat{\mathbf{Q}}_i$ is the SLERP rotation input. FK is a forward kinematic operator that, given skeleton offset and joint rotations, returns the joint positions. $\mathbf{P}_i = \mathrm{FK}(\mathbf{S}, \mathbf{Q}_i)$ are the joint positions of the ground truth dance sequence.

5 IMPLEMENTATION

Our system's design and the methodological approach we have adopted are versatile, allowing for implementation as either a standalone unit or within a larger online framework. In this paper, we offer a reference implementation of our system as a web application specifically designed for a standalone kiosk shown in Figure 2.

5.1 Hardware Design

Our kiosk setup is designed to deliver a realistic and immersive experience, featuring a 65-inch portrait monitor for displaying life-size dancing Bitmoji characters. This setup caters to a wide range of music sources, allowing users to play their choice of music from smartphones, tablets, MP3 players, or even online web players. An integral part of this setup is a QR code scanner [4], which lets users bring their personalized Bitmoji avatars into the dance animation system, adding a layer of personalization to the experience. We leverage The computational processing is handled by an Apple M2 Ultra Mac Studio, which synthesizes the 3D dance animations in realtime that are synchronized with the music.

Figure 2: *DanceCraft Kiosk:* The kiosk features an iPad for user to play music. The avatar dances in response to the music on a life-sized portrait monitor. Users can cast their personal 3D Bitmoji by presenting a QR code to the QR code scanner.

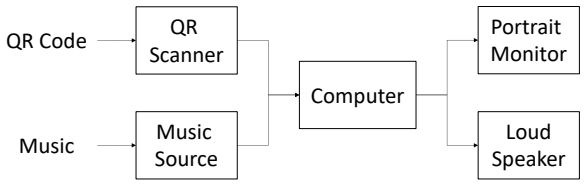


Figure 3: Hardware setup topology diagram.

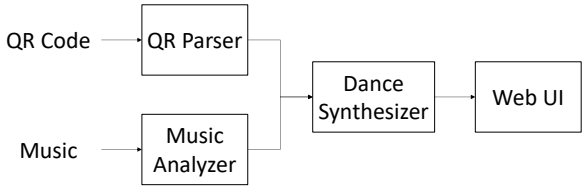


Figure 4: High level software architecture

5.2 Software Design

Our system's versatility is enhanced by its implementation as a web application. The software topology diagram, as shown in Figure 4, provides a detailed overview of our system's architecture. Key components like `Music Analyzer` and `QR Parser` are implemented

server-side, handling the backend processes. On the client side, we have developed the `Dance Synthesizer` engine as a JavaScript application, complemented by a WebGL presentation layer for visually rendering the animations.

The `Music Analyzer` processes the incoming music stream using a moving window approach. Based on empirical analysis, we have determined an appropriate window size of 3 seconds and a moving increment of 0.3 seconds. Within each window, the `Music Analyzer` extracts key musical characteristics, as detailed in Section 4.2. This process allows for the extraction of pertinent musical features such as genre, tempo, and energy. The extracted data is then relayed in real-time to the `Dance Synthesizer` engine.

For users with Bitmoji avatars, the integration is seamless. They can retrieve a QR code encoding the URL of their Bitmoji through Snapchat. Scanning this code with the kiosk's QR code scanner enables their personalized avatar to be projected into the animation system. It is important to highlight that all Bitmojis from different users share a common skeletal structure. This unified skeleton enables us to seamlessly switch between different Bitmojis in real-time according to the user's request. However, as stated in 4.3.3, Bitmojis can have different body shapes and sizes – from skinny to obese mesh. The metadata embedded in the retreived Bitmoji model aids the dance synthesizer in switching to the appropriate set of dance animations to prevent interpenetration.

6 EVALUATION

This section presents the quantitative evaluations of our method, comparisons to baseline and prior works, and ablation studies. We also conduct user studies to assess the proposed system qualitatively.

6.1 Quantitative Evaluation

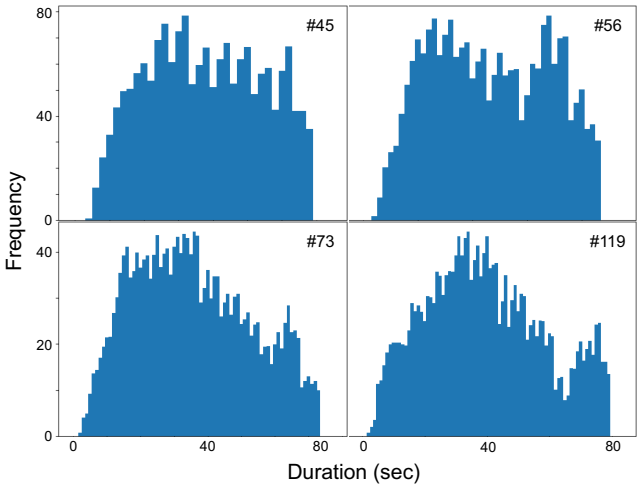


Figure 5: *Dance track coverage histogram.* We show the coverage histograms for four randomly selected dance tracks from our database. It demonstrates our system's capability to fully utilize each track for dance synthesis.

Table 1: *Comparison on LAFAN1 for motion in-between task (30 frames).* The best performance in each category is indicated in bold and the second-best is highlighted in cyan.

Length = 30	L2P ↓	L2Q ↓	NPSS ↓
Zero-Velocity	6.60	1.51	0.2318
Interpolation	2.32	0.98	0.2013
ERD-QV [26]	1.28	0.69	0.1328
SSMCT [18]	1.10	0.61	0.1222
CMIB [38]	1.19	0.59	0.1415
Δ-Interp [64]	**1.00**	0.57	0.1217
Ours	1.04	**0.54**	**0.1213**

Dance track coverage. A critical aspect of evaluating the performance of our proposed system is the assessment of how comprehensively individual dance tracks within our database are utilized. An ideal functioning of the system, in line with our dense motion graph approach described in Section 4.3, would involve traversing the full range of all dance tracks in our collection.

To quantitatively evaluate this aspect, we conducted a series of tests where the system was fed input songs across a spectrum of tempos, ranging from 80 BPM to 160 BPM, in increments of 5 BPM. For each tempo level, we initiated 1000 transitions and observed the extent of coverage for each dance track. The coverage results for four dance tracks, selected at random, are illustrated in Figure 5. These findings demonstrate that our system consistently covers the entire span of each selected dance track, showcasing its ability to effectively utilize the entire breadth of the dance database. This extensive coverage is indicative of the system's capacity to provide a diverse and comprehensive dance experience, adapting to a wide range of musical tempos and styles.

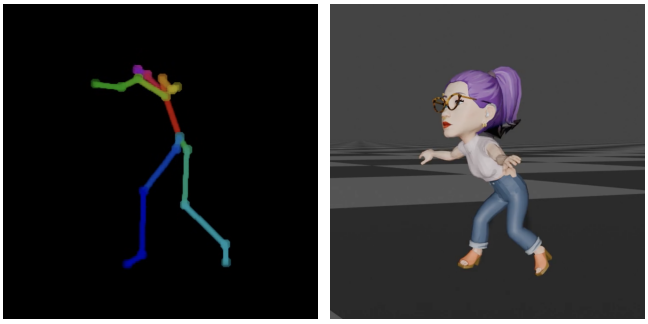
Motion in-betweening methods comparison. To assess the efficacy of our `Dance Synthesizer`, particularly $\mathcal{G}_T$, the real-time generation of 30-frame motion in-betweening, we conducted a comprehensive evaluation using the LAFAN1 dataset [26]. Our evaluation employed three metrics: i) L2 distances of global positions (L2P)[26], ii) L2 distances of global rotations (L2Q)[26], and iii) Normalized Power Spectrum Similarity (NPSS) [22].

Our $\mathcal{G}_T$ was compared against various baselines and existing methods. The zero-velocity baseline approach involves replacing the intervening frames with the last frame of the source motion. The interpolation baseline utilizes linear interpolation (LERP on the global root position and SLERP on the rotation of each joint) between the final frame of the source motion and the initial frame of the target motion. Additionally, we compared our method with several prior techniques, including ERD-QV [26], SSMCT [18], CMIB [38], and Δ-Interp [64].

The results of this quantitative evaluation, presented in Table 1, demonstrate the robustness of our approach. Our method not only surpasses previous state-of-the-art (STOA) methods like ERD-QV, SSMCT, and CMIB in performance but also competes favorably with concurrent works, namely Δ-Interp. This indicates the effectiveness of our $\mathcal{G}_T$ in accurately and realistically generating intermediate

Table 2: *Ablation study.* We compared the performance of our proposed $\mathcal{G}_T$ with and without the transformer component. The best score under each metric is emphasized in bold.

Length = 30	L2P ↓	L2Q ↓	NPSS ↓
Ours w/o Trans	1.31	0.73	0.1337
Ours	**1.04**	**0.54**	**0.1213**



(a) Bailando++ output **(b) Retargeted Bitmoji**

Figure 6: *Data preparation for user studies.* We retargeted the output from Bailando++, originally in SMPL skeleton format, to the Bitmoji format. We then associated a random Bitmoji character with the skeleton, ensuring a fair comparison between the outputs of Bailando++ and our proposed system.

dance frames in real-time, a crucial capability for the system's overall functionality.

Ablation study. To validate the impact of the transformer component in our $\mathcal{G}_T$, we conducted an ablation study. This involved removing the transformer component from our model for comparative analysis. The modified version of our model without the transformer is denoted as "Ours w/o Trans", and we juxtaposed its performance against our complete model, denoted as "Ours".

In the absence of the transformer, our model adopts a configuration akin to a denoising autoencoder, aligning with the structural outline provided in Section 4.3. The comparative results, detailed in Table 2, underscore a significant performance discrepancy between the two versions. The findings from this ablation study highlight the critical role of the transformer component in enhancing the system's capability to accurately and dynamically generate dance motions, thereby validating its inclusion in our $\mathcal{G}_T$.

6.2 Qualitative User Studies

Quantitative evaluations highlight our $\mathcal{G}_T$'s in-betweening abilities, but they only partly represent the goal of our full system, `Dance Synthesizer`, which is to generate real-time, music-reactive improvisational dances. GrooveNet [8], the most closely related work, also focuses on real-time dance generation but struggles in generalizing beyond its training data. Hence, for a more effective comparison, we included Bailando++ [79], an autoregressive method known for its qualitative strength in generating 3D dance movements based on music.

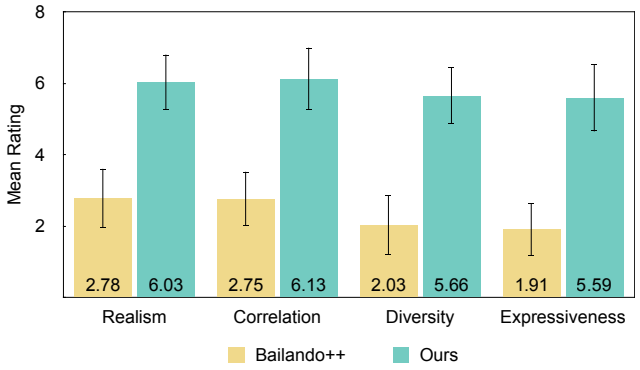


Figure 7: *Quality of dance comparison.* We conducted a blind comparison between dances generated by Bailando++ and our method. The results reveal a strong preference among participants for our system's dance quality.

Participants. We recruited 32 participants, comprising 19 males and 13 females, aged between 18 and 37. None of the participants had previous experience with 3D dance systems.

User Study 1: Quality of dance.

Data preparation. Bailando++ is *not* a real-time method, as it conditions dance generation on the entirety of the input music. To ensure a fair comparison in our user studies, we prepared data as follows:

(1) Since Bailando++ is trained on AIST++ dataset [55], we randomly selected 20 music clips from its test set to generate dance movements for each.
(2) Bailando++ generates dances in SMPL format [57]. We retargeted the outputs to the Bitmoji skeleton format, and assigned random Bitmoji characters to the skeletons. The resulting dance sequences were then converted into videos, as shown in Figure 6a and Figure 6b.
(3) Using the same 20 music clips, our system generated corresponding dance sequences in real-time. These sequences were recorded into videos for comparison.

Procedure and results. To compare dance quality, participants engaged in a blind test viewing 20 videos from either Bailando++ or our system. They rated each video on a scale from 1 to 7 across four dimensions: dance realism, music-dance correlation, dance diversity, and movement expressiveness. Participants then repeated the procedure with the other system. To counter order bias, the study sequence was counterbalanced. Figure 7 presents the average user ratings, indicating a marked preference for our proposed system.

User Study 2: User experience. Next, we evaluated participants' user experience with both systems. For ours, participants freely chose music, paused, and switched tracks, as outlined in Section 3. With Bailando++, participants used the same iPad kiosk to choose music. Upon selection, they were asked to run an all-in-one script for music download and processing, output generation and processing, and dance video creation. This process, though automated to the best extent, still took about 5 minutes or more. Participants rated their experience on enjoyment, engagement, ease of use, and responsiveness on a 1-7 scale. They then repeated the process with

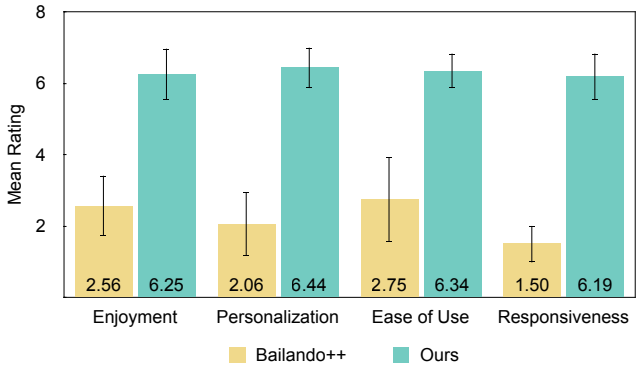


Figure 8: *User experience comparison.* In a blind comparison of Bailando++ and our method, participants significantly preferred our system's user experience, highlighting its superior engagement and satisfaction.

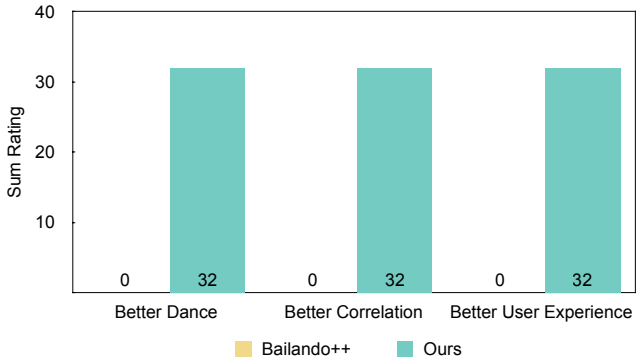


Figure 9: *Overall preference of participants.* In the concluding phase of our study, participants chose their overall preferred system, unanimously favoring the experience and capabilities of our system.

the other system. The identity of the systems was not disclosed to the participants. A blind comparison was carried out in a counterbalanced order to minimize bias. The mean user ratings for both systems, as depicted in Figure 8, clearly demonstrate a strong preference for our proposed system.

User Study 3: Overall preference. Upon completing the blind interaction with both systems, participants were asked to express their overall preferences through three categories, including overall dance quality, music-dance correlation, and user experience. Figure 9 shows the results of this final study. Despite not being informed about the identities of the systems, the participants unanimously favored our proposed system across all the categories. This unanimous preference underscores the effectiveness of our system in delivering a superior user experience, characterized by the combination of dance quality and music-dance synchronization.

Positive feedback. The participants' response to our proposed system was overwhelmingly positive. A notable observation was the universal expression of enjoyment, as evidenced by the smiles on the faces of all participants while interacting with the system. 27 out of 32 participants expressed astonishment and delight at seeing their Bitmoji dance in sync with their chosen music on the large portrait TV. Additionally, 23 participants actively joined in the experience, mirroring the dance movements of their virtual counterparts. Furthermore, 12 participants expressed interest in integrating our system into VR dance parties. Notable feedback included Participant #8 returning for a second interaction with their daughter and Participant #14's intrigue about the system's potential adaptability to complex musical pieces like Broadway shows. The overall positive user rating and subjective feedback highlight the system's practicality and appeal for various applications, from personal use to social and virtual events.

User suggestions. Participants also provided constructive suggestions for improvement. Many requested a wider range of dance styles reflecting culture diversity, such as traditional Chinese ribbon dance and Japanese Awa Odori.Another common suggestion was mobile device compatibility, enhancing accessibility and convenience. Additionally, over half of the participants envisioned integrating the system with AR/VR headsets, seeing it as an enriching addition to immersive virtual environments. Participant #27's innovative idea involved creating a physical gadget reflecting their 3D Bitmoji, capable of reacting to music, inspired by our virtual system. This suggests new, tangible applications of our technology.

7 LIMITATIONS AND FUTURE WORK

Our emphasis on improvisational dances comes at the cost of choreographic accuracy and semantic nuances, which might be perceived as a drawback by trained dancers. Another limitation lies in our current dance routine retrieval system, which operates on a basic lookup table approach based on music genre, tempo, and energy. Future enhancements aim to evolve this into a more sophisticated two-tower embedding model [92] with predictive modeling, capable of not only retrieving semantically relevant dances but also ensuring synchronization with musical beats. Currently, another drawback of our system is that it is not computationally efficient on low-end computers. We would like optimize the method so that it can even run on mobile phones.

8 CONCLUSION

In this paper, we introduced a production-ready, real-time system adept at generating realistic, expressive, and captivating improvisational dances in response to music. Our approach overcomes the limitations of both data-driven graph-based and deep learning approaches by implementing a novel hybrid method. We also unveiled a comprehensive dance dataset encompassing a diverse array of musical genres, tempos, and energy levels. Our system emphasizes user interactivity and personalization, creating a deeply engaging experience. Significantly, user study results show a marked preference for our system's generated dance motions among untrained users. Finally, our implementation of the system with web technologies enables versatile deployment options. We can set up the dance generator as a standalone kiosk or integrate it into larger virtual concert environments.

ACKNOWLEDGMENTS

To Robert, for the bagels and explaining CMYK and color spaces.

REFERENCES

[1] [n. d.]. Bitmoji. https://www.bitmoji.com/. Accessed: 2024-01-01.

[2] [n. d.]. Happy 15th Birthday, Bitmoji. https://newsroom.snap.com/happy-birthday-bitmoji/. Accessed: 2024-01-01.

[3] [n. d.]. Jingle Punks. https://www.jinglepunks.com/. Accessed: 2024-01-01.

[4] [n. d.]. Newland FM3080 Hind. https://https://www.newland-id.com/en/products/stationary-scanners/fm3080-hind/. Accessed: 2024-01-01.

[5] Kfir Aberman, Peizhuo Li, Dani Lischinski, Olga Sorkine-Hornung, Daniel Cohen-Or, and Baoquan Chen. 2020. Skeleton-Aware Networks for Deep Motion Retargeting. *ACM Transactions on Graphics (TOG)* 39, 4 (2020), 62.

[6] Emre Aksan, Manuel Kaufmann, Peng Cao, and Otmar Hilliges. 2021. A Spatio-temporal Transformer for 3D Human Motion Prediction. arXiv:2004.08692 [cs.CV]

[7] Emre Aksan, Manuel Kaufmann, and Otmar Hilliges. 2019. Structured Prediction Helps 3D Human Motion Modelling. arXiv:1910.09070 [cs.CV]

[8] Omid Alemi and Philippe Pasquier. 2017. GrooveNet : Real-Time Music-Driven Dance Movement Generation using Artificial Neural Networks. https://api.semanticscholar.org/CorpusID:52062683

[9] Okan Arikan and D. A. Forsyth. 2002. Interactive Motion Generation from Examples. *ACM Trans. Graph.* 21, 3 (July 2002), 483–490. https://doi.org/10.1145/566654.566606

[10] Alexander Berman and Valencia James. 2015. Kinetic Imaginations: Exploring the Possibilities of Combining AI and Dance. In *Proceedings of the 24th International Conference on Artificial Intelligence* (Buenos Aires, Argentina) *(Ijcai'15)*. AAAI Press, 2431–2437.

[11] R Bowden. 2000. Learning Statistical Models of Human Motion.

[12] Matthew Brand and Aaron Hertzmann. 2000. Style Machines. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques (Siggraph '00)*. ACM Press/Addison-Wesley Publishing Co., Usa, 183–192. https://doi.org/10.1145/344779.344865

[13] Judith Bütepage, Michael Black, Danica Kragic, and Hedvig Kjellström. 2017. Deep representation learning for human motion prediction and classification. arXiv:1702.07486 [cs.CV]

[14] Katy Carey, Aidan Moran, and Brendan Rooney. 2019. Learning Choreography: An Investigation of Motor Imagery, Attentional Effort, and Expertise in Modern Dance. *Frontiers in Psychology* 10 (March 2019). https://doi.org/10.3389/fpsyg.2019.00422

[15] Jinxiang Chai and Jessica K. Hodgins. 2007. Constraint-Based Motion Optimization Using a Statistical Dynamic Model. *ACM Trans. Graph.* 26, 3 (July 2007), 8–es. https://doi.org/10.1145/1276377.1276387

[16] Cmu. 2010. http://mocap.cs.cmu.edu

[17] Luka Crnkovic-Friis and Louise Crnkovic-Friis. 2016. Generative Choreography using Deep Learning. arXiv:1605.06921 [cs.AI]

[18] Yinglin Duan, Tianyang Shi, Zhengxia Zou, Yenan Lin, Zhehui Qian, Bohan Zhang, and Yi Yuan. 2021. Single-Shot Motion Completion with Transformer. arXiv:2103.00776 [cs.CV]

[19] João Pedro Moreira Ferreira, Thiago M. Coutinho, Thiago L. Gomes, José F. Neto, Rafael Azevedo, Renato Martins, and Erickson R. Nascimento. 2020. Learning to dance: A graph convolutional adversarial network to generate realistic dance motions from audio. *CoRR* abs/2011.12999 (2020). arXiv:2011.12999 https://arxiv.org/abs/2011.12999

[20] Katerina Fragkiadaki, Sergey Levine, Panna Felsen, and Jitendra Malik. 2015. Recurrent Network Models for Human Dynamics. arXiv:1508.00271 [cs.CV]

[21] Partha Ghosh, Jie Song, Emre Aksan, and Otmar Hilliges. 2017. Learning Human Motion Models for Long-term Predictions. arXiv:1704.02827 [cs.CV]

[22] Anand Gopalakrishnan, Ankur Mali, Dan Kifer, C. Lee Giles, and Alexander G. Ororbia. 2019. A Neural Temporal Model for Human Motion Prediction. arXiv:1809.03036 [cs.CV]

[23] Liang-Yan Gui, Yu-Xiong Wang, Xiaodan Liang, and José M. F. Moura. 2018. Adversarial Geometry-Aware Human Motion Prediction. In *Computer Vision – ECCV 2018*, Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss (Eds.). Springer International Publishing, Cham, 823–842.

[24] Chuan Guo, Yuxuan Mu, Muhammad Gohar Javed, Sen Wang, and Li Cheng. 2023. MoMask: Generative Masked Modeling of 3D Human Motions. (2023). arXiv:2312.00063 [cs.CV]

[25] Félix G. Harvey and Christopher Pal. 2021. Recurrent Transition Networks for Character Locomotion. arXiv:1810.02363 [cs.GR]

[26] Félix G. Harvey, Mike Yurick, Derek Nowrouzezahrai, and Christopher Pal. 2020. Robust Motion In-Betweening. *ACM Trans. Graph.* 39, 4, Article 60 (Aug. 2020), 12 pages. https://doi.org/10.1145/3386569.3392480

[27] Mojtaba Heydari, Frank Cwitkowitz, and Zhiyao Duan. 2021. BeatNet: CRNN and Particle Filtering for Online Joint Beat Downbeat and Meter Tracking. In *22th International Society for Music Information Retrieval Conference, ISMIR*.

[28] Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising Diffusion Probabilistic Models. *arXiv preprint arxiv:2006.11239* (2020).

[29] Daniel Holden, Jun Saito, and Taku Komura. 2016. A Deep Learning Framework for Character Motion Synthesis and Editing. *ACM Trans. Graph.* 35, 4, Article 138 (July 2016), 11 pages. https://doi.org/10.1145/2897824.2925975

[30] Daniel Holden, Jun Saito, Taku Komura, and Thomas Joyce. 2015. Learning Motion Manifolds with Convolutional Autoencoders. In *SIGGRAPH Asia 2015 Technical Briefs* (Kobe, Japan) *(Sa '15)*. Association for Computing Machinery, New York, NY, USA, Article 18, 4 pages. https://doi.org/10.1145/2820903.2820918

[31] Ruozi Huang, Huang Hu, Wei Wu, Kei Sawada, Mi Zhang, and Daxin Jiang. 2023. Dance Revolution: Long-Term Dance Generation with Music via Curriculum Learning. arXiv:2006.06119 [cs.CV]

[32] Adobe Systems Inc. 2018. https://www.mixamo.com

[33] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. 2014. Human3.6M: Large Scale Datasets and Predictive Methods for 3D Human Sensing in Natural Environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36, 7 (2014), 1325–1339. https://doi.org/10.1109/tpami.2013.248

[34] Ashesh Jain, Amir R. Zamir, Silvio Savarese, and Ashutosh Saxena. 2016. Structural-RNN: Deep Learning on Spatio-Temporal Graphs. arXiv:1511.05298 [cs.CV]

[35] Chen Kang, Zhipeng Tan, Jin Lei, Song-Hai Zhang, Yuan-Chen Guo, Weidong Zhang, and Shi-Min Hu. 2021. ChoreoMaster: Choreography-Oriented Music-Driven Dance Synthesis. *ACM Transactions on Graphics (TOG)* 40, 4 (2021).

[36] Hsuan-Kai Kao and Li Su. 2020. Temporally Guided Music-to-Body-Movement Generation. In *Proceedings of the 28th ACM International Conference on Multimedia (MM '20)*. Acm. https://doi.org/10.1145/3394171.3413848

[37] Manuel Kaufmann, Emre Aksan, Jie Song, Fabrizio Pece, Remo Ziegler, and Otmar Hilliges. 2020. Convolutional Autoencoders for Human Motion Infilling. In *2020 International Conference on 3D Vision (3DV)*. Ieee. https://doi.org/10.1109/3dv50981.2020.00102

[38] Jihoon Kim, Taehyun Byun, Seungyoun Shin, Jungdam Won, and Sungjoon Choi. 2022. Conditional Motion In-betweening. *Pattern Recognition* (2022), 108894. https://doi.org/10.1016/j.patcog.2022.108894

[39] Jihoon Kim, Jiseob Kim, and Sungjoon Choi. 2022. Flame: Free-form language-based motion synthesis & editing. *arXiv preprint arXiv:2209.00349* (2022).

[40] Jae Woo Kim, Hesham Fouad, and James K. Hahn. 2006. Making Them Dance. In *AAAI Fall Symposium: Aurally Informed Performance*. https://api.semanticscholar.org/CorpusID:18861896

[41] Tae-hoon Kim, Sang Il Park, and Sung Yong Shin. 2003. Rhythmic-Motion Synthesis Based on Motion-Beat Analysis. *ACM Trans. Graph.* 22, 3 (July 2003), 392–401. https://doi.org/10.1145/882262.882283

[42] Lucas Kovar, Michael Gleicher, and Frédéric Pighin. 2008. Motion graphs. *ACM SIGGRAPH 2008 classes* (2008). https://doi.org/10.1145/1401132.1401202

[43] Lucas Kovar, Michael Gleicher, and Frédéric H. Pighin. 2002. Motion graphs. In *International Conference on Computer Graphics and Interactive Techniques*. https://api.semanticscholar.org/CorpusID:2063215

[44] Hsu kuang Chiu, Ehsan Adeli, Borui Wang, De-An Huang, and Juan Carlos Niebles. 2018. Action-Agnostic Human Pose Forecasting. arXiv:1810.09676 [cs.CV]

[45] Alexis Lamouret and Michiel van de Panne. 1996. Motion Synthesis By Example. In *Computer Animation and Simulation '96*, Ronan Boulic and Gerard Hégron (Eds.). Springer Vienna, Vienna, 199–212.

[46] Manfred Lau, Ziv Bar-Joseph, and James Kuffner. 2009. Modeling Spatial and Temporal Variation in Motion Data. *ACM Trans. Graph.* 28, 5 (Dec. 2009), 1–10. https://doi.org/10.1145/1618452.1618517

[47] Hsin-Ying Lee, Xiaodong Yang, Ming-Yu Liu, Ting-Chun Wang, Yu-Ding Lu, Ming-Hsuan Yang, and Jan Kautz. 2019. Dancing to Music. arXiv:1911.02001 [cs.CV]

[48] Jehee Lee, Jinxiang Chai, Paul S. A. Reitsma, Jessica K. Hodgins, and Nancy S. Pollard. 2002. Interactive Control of Avatars Animated with Human Motion Data. 21, 3 (July 2002), 491–500. https://doi.org/10.1145/566654.566607

[49] Juheon Lee, Seohyun Kim, and Kyogu Lee. 2018. Listen to Dance: Music-driven choreography generation using Autoregressive Encoder-Decoder Network. arXiv:1811.00818 [cs.MM]

[50] Jehee Lee and Sung Yong Shin. 1999. A Hierarchical Approach to Interactive Motion Editing for Human-like Figures. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques (Siggraph '99)*. ACM Press/Addison-Wesley Publishing Co., Usa, 39–48. https://doi.org/10.1145/311535.311539

[51] Minho Lee, Kyogu Lee, and Jaeheung Park. 2013. Music similarity-based approach to generating dance motion sequence. *Multimedia Tools and Applications* 62 (02 2013). https://doi.org/10.1007/s11042-012-1288-5

[52] Andreas M. Lehrmann, Peter V. Gehler, and Sebastian Nowozin. 2014. Efficient Nonlinear Markov Models for Human Motion. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*. 1314–1321. https://doi.org/10.1109/cvpr.2014.171

[53] Buyu Li, Yongchi Zhao, Shi Zhelun, and Lu Sheng. 2022. Danceformer: Music conditioned 3d dance generation with parametric motion transformer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 36. 1272–1279.

[54] Jiaman Li, Yihang Yin, Hang Chu, Yi Zhou, Tingwu Wang, Sanja Fidler, and Hao Li. 2020. Learning to Generate Diverse Dance Motions with Transformer. arXiv:2008.08171 [cs.CV]

[55] Ruilong Li, Shan Yang, David A. Ross, and Angjoo Kanazawa. 2021. AI Choreographer: Music Conditioned 3D Dance Generation with AIST++.

[56] Weiyu Li, Xuelin Chen, Peizhuo Li, Olga Sorkine-Hornung, and Baoquan Chen. 2023. Example-Based Motion Synthesis via Generative Motion Matching. *ACM Transactions on Graphics (TOG)* 42, 4, Article 94 (2023). https://doi.org/10.1145/3592395

[57] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. 2015. SMPL: A Skinned Multi-Person Linear Model. *ACM Trans. Graphics (Proc. SIGGRAPH Asia)* 34, 6 (Oct. 2015), 248:1–248:16.

[58] Mônica m. Ribeiro and Agar Fonseca. 2011. The empathy and the structuring sharing modes of movement sequences in the improvisation of contemporary dance. *Research in Dance Education* 12, 2 (2011), 71–85. https://doi.org/10.1080/14647893.2011.575220

[59] Naureen Mahmood, Nima Ghorbani, Nikolaus F. Troje, Gerard Pons-Moll, and Michael J. Black. 2019. AMASS: Archive of Motion Capture as Surface Shapes. In *International Conference on Computer Vision*. 5442–5451.

[60] Julieta Martinez, Michael J. Black, and Javier Romero. 2017. On human motion prediction using recurrent neural networks. arXiv:1705.02445 [cs.CV]

[61] Jianyuan Min and Jinxiang Chai. 2012. Motion Graphs++: A Compact Generative Model for Semantic Motion Analysis and Synthesis. *ACM Trans. Graph.* 31, 6, Article 153 (Nov. 2012), 12 pages. https://doi.org/10.1145/2366145.2366172

[62] Tomohiko Mukai and Shigeru Kuriyama. 2005. Geostatistical Motion Interpolation. *ACM Trans. Graph.* 24, 3 (July 2005), 1062–1070. https://doi.org/10.1145/1073204.1073313

[63] Yuko Nakano and Takeshi Okada. 2012. Process of Improvisational Contemporary Dance. In *34th Annual Meeting of the Cognitive Science Society*.

[64] Boris N. Oreshkin, Antonios Valkanas, Félix G. Harvey, Louis-Simon Ménard, Florent Bocquelet, and Mark J. Coates. 2022. Motion Inbetweening via Deep Δ-Interpolator. arXiv:2201.06701 [cs.LG]

[65] Dario Pavllo, David Grangier, and Michael Auli. 2018. QuaterNet: A Quaternion-based Recurrent Model for Human Motion. arXiv:1805.06485 [cs.CV]

[66] Vladimir Pavlovic, James M. Rehg, and John MacCormick. 2000. Learning Switching Linear Models of Human Motion. In *Proceedings of the 13th International Conference on Neural Information Processing Systems* (Denver, CO) *(Nips'00)*. MIT Press, Cambridge, MA, USA, 942–948.

[67] Steve Paxton. 1975. Contact Improvisation. *The Drama Review: TDR* 19, 1 (1975), 40–42. http://www.jstor.org/stable/1144967

[68] M. Plakal and D. Ellis. 2020. YAMNet. https://github.com/tensorflow/models/tree/master/research/audioset/yamnet.

[69] Jia Qin, Youyi Zheng, and Kun Zhou. 2022. Motion In-Between via Two-Stage Transformers. *ACM Trans. Graph.* 41, 6, Article 184 (Nov. 2022), 16 pages. https://doi.org/10.1145/3550454.3555454

[70] Sigal Raab, Inbal Leibovitch, Guy Tevet, Moab Arar, Amit H. Bermano, and Daniel Cohen-Or. 2023. Single Motion Diffusion. arXiv:2302.05905 [cs.CV]

[71] Rokoko. 2023. Capture your body's motion in real-time with Smartsuit Pro II. https://www.rokoko.com/products/smartsuit-pro

[72] C. Rose, M.F. Cohen, and B. Bodenheimer. 1998. Verbs and adverbs: multidimensional motion interpolation. *IEEE Computer Graphics and Applications* 18, 5 (1998), 32–40. https://doi.org/10.1109/38.708559

[73] Alejandro Hernandez Ruiz, Juergen Gall, and Francesc Moreno-Noguer. 2019. Human Motion Prediction via Spatio-Temporal Inpainting. arXiv:1812.05478 [cs.CV]

[74] Alla Safonova and Jessica K. Hodgins. 2007. Construction and Optimal Search of Interpolated Motion Graphs. *ACM Trans. Graph.* 26, 3 (July 2007), 106–es. https://doi.org/10.1145/1276377.1276510

[75] Ken'ichi Sasaki. 1995. *Bigaku Jiten = Dictionary of Aesthetics*. Tokyo Daigaku Shuppankai.

[76] Sfu. 2017. https://mocap.cs.sfu.ca

[77] Takaaki Shiratori, Atsushi Nakazawa, and Katsushi Ikeuchi. 2006. Dancing-to-Music Character Animation. *Computer Graphics Forum* 25, 3 (2006), 449–458. https://doi.org/10.1111/j.1467-8659.2006.00964.x arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1467-8659.2006.00964.x

[78] Li Siyao, Weijiang Yu, Tianpei Gu, Chunze Lin, Quan Wang, Chen Qian, Chen Change Loy, and Ziwei Liu. 2022. Bailando: 3D dance generation via Actor-Critic GPT with Choreographic Memory. In *Cvpr*.

[79] Li Siyao, Weijiang Yu, Tianpei Gu, Chunze Lin, Quan Wang, Chen Qian, Chen Change Loy, and Ziwei Liu. 2023. Bailando++: 3D Dance GPT With Choreographic Memory. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45, 12 (2023), 14192–14207. https://doi.org/10.1109/tpami.2023.3319435

[80] Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. 2015. Deep Unsupervised Learning using Nonequilibrium Thermodynamics. arXiv:1503.03585 [cs.LG]

[81] G. Stevens. 1975. *Psychophysics: Introduction to its perceptual, neural, and social prospects*. John Wiley & Sons.

[82] Guofei Sun, Yongkang Wong, Zhiyong Cheng, Mohan S. Kankanhalli, Weidong Geng, and Xiangdong Li. 2021. DeepDance: Music-to-Dance Motion Choreography With Adversarial Learning. *IEEE Transactions on Multimedia* 23 (2021), 497–509. https://doi.org/10.1109/tmm.2020.2981989

[83] L.M. Tanco and A. Hilton. 2000. Realistic synthesis of novel human movements from a database of motion capture examples. In *Proceedings Workshop on Human Motion*. 137–142. https://doi.org/10.1109/humo.2000.897383

[84] Taoran Tang, Jia Jia, and Hanyang Mao. 2018. Dance with Melody: An LSTM-Autoencoder Approach to Music-Oriented Dance Synthesis. In *Proceedings of the 26th ACM International Conference on Multimedia* (Seoul, Republic of Korea) *(MM '18)*. Association for Computing Machinery, New York, NY, USA, 1598–1606. https://doi.org/10.1145/3240508.3240526

[85] Xiangjun Tang, He Wang, Bo Hu, Xu Gong, Ruifan Yi, Qilong Kou, and Xiaogang Jin. 2022. Real-time controllable motion transition for characters. *ACM Transactions on Graphics* 41, 4 (July 2022), 1–10. https://doi.org/10.1145/3528223.3530090

[86] Guy Tevet, Sigal Raab, Brian Gordon, Yoni Shafir, Daniel Cohen-or, and Amit Haim Bermano. 2023. Human Motion Diffusion Model. In *The Eleventh International Conference on Learning Representations*. https://openreview.net/forum?id=SJ1kSyO2jwu

[87] Jonathan Tseng, Rodrigo Castellon, and C Karen Liu. 2022. EDGE: Editable Dance Generation From Music. *arXiv preprint arXiv:2211.10658* (2022).

[88] Shuhei Tsuchida, Satoru Fukayama, Masahiro Hamasaki, and Masataka Goto. 2019. AIST Dance Video Database: Multi-genre, Multi-dancer, and Multi-camera Database for Dance Information Processing. In *Proceedings of the 20th International Society for Music Information Retrieval Conference, ISMIR 2019*. Delft, Netherlands.

[89] Borui Wang, Ehsan Adeli, Hsu kuang Chiu, De-An Huang, and Juan Carlos Niebles. 2019. Imitation Learning for Human Pose Prediction. arXiv:1909.03449 [cs.CV]

[90] Nelson Yalta, Shinji Watanabe, Kazuhiro Nakadai, and Tetsuya Ogata. 2019. Weakly Supervised Deep Recurrent Neural Networks for Basic Dance Step Generation. arXiv:1807.01126 [cs.LG]

[91] Zijie Ye, Haozhe Wu, Jia Jia, Yaohua Bu, Wei Chen, Fanbo Meng, and Yanfeng Wang. 2020. ChoreoNet: Towards Music to Dance Synthesis with Choreographic Action Unit. In *Proceedings of the 28th ACM International Conference on Multimedia*. Acm. https://doi.org/10.1145/3394171.3414005

[92] Xinyang Yi, Ji Yang, Lichan Hong, Derek Zhiyuan Cheng, Lukasz Heldt, Aditee Ajit Kumthekar, Zhe Zhao, Li Wei, and Ed Chi (Eds.). 2019. *Sampling-Bias-Corrected Neural Modeling for Large Corpus Item Recommendations*.

[93] Jiaxu Zhang, Junwu Weng, Di Kang, Fang Zhao, Shaoli Huang, Xuefei Zhe, Linchao Bao, Ying Shan, Jue Wang, and Zhigang Tu. 2023. Skinned Motion Retargeting with Residual Perception of Motion Semantics & Geometry. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 13864–13872.

[94] Mingyuan Zhang, Zhongang Cai, Liang Pan, Fangzhou Hong, Xinying Guo, Lei Yang, and Ziwei Liu. 2022. MotionDiffuse: Text-Driven Human Motion Generation with Diffusion Model. *arXiv preprint arXiv:2208.15001* (2022).

[95] Yi Zhou, Jingwan Lu, Connelly Barnes, Jimei Yang, Sitao Xiang, and Hao li. 2020. Generative Tweening: Long-term Inbetweening of 3D Human Motions. arXiv:2005.08891 [cs.CV]

[96] Wenlin Zhuang, Congyi Wang, Jinxiang Chai, Yangang Wang, Ming Shao, and Siyu Xia. 2022. Music2Dance: DanceNet for Music-Driven Dance Generation. 18, 2, Article 65 (Feb. 2022), 21 pages. https://doi.org/10.1145/3485664

Received 20 February 2007; revised 12 March 2009; accepted 5 June 2009