DuoDance:
Audio-driven generation of partner dancing animations using diffusion models

Jake Greenshields
1600702

May 2024

Supervised by
Dr. Ronald Poppe
Dr. Zerrin Yumak

Department of Information and Computing Sciences
Utrecht University
Game and Media Technology

Contents

1 Introduction

The study of human motion generation has been an active area of research for decades and combines computer graphics, computer vision, and machine learning. Modeling human motion has practical applications in a multitude of fields including self-driving cars with pedestrian motion prediction, security and surveillance, human-computer interaction, human behavior analysis, and many more. Human motion synthesis has seen increased use within video games and virtual experiences. In order to create realistic and persuasive animations, techniques must be able to model the complex human motion manifold. One of the most challenging representations of human motion is that of dance. Different dances encompass a wide variety of styles and genres, each with its own rhythm and unique movements. These movements can be large and involve every aspect of the performer's body meaning interdependencies between different body parts should be considered. Some dances involve two or more individuals coordinating their movements, communicating through physical contact, and paying attention to their surroundings. Finally, conditioning this generation of motion based on audio adds another layer of complexity to ensure the movements are coherent and representative of the music.

There have been numerous attempts to synthesize motion from music for a single dancer [1–16]. However, work involving multiple dancers is limited [17, 18] and doesn't take into consideration any interactions. In partner dancing, it is common to have a leader and a follower. The leader is responsible for guiding the dance and choosing what moves to perform. They do this by giving the follower subtle signals both physically and visually. What moves the leader decides to lead requires an awareness of the follower's position meaning both dancers rely on physical and visual cues to determine their movement. Recent advancements in text-to-motion diffusion models to produce diverse interactive motion [19, 20] provide a suitable starting point for the task of audio-driven partner dance synthesis.

1.1 Research Questions

This research project aims to tackle the underlooked area of partner dancing motion synthesis. More specifically, our goal is to adapt state-of-the-art (SOTA) diffusion models to generate realistic motion for two interacting dancers when conditioned on audio input. This motion should be physically plausible and representative of captured motion associated with the audio. The work will be quantitatively assessed using SOTA measuring techniques and qualitatively assessed through the use of perceptual studies. The subsequent research questions are formulated to guide the generation and evaluation of partner dancing motion synthesis using SOTA diffusion models conditioned on audio input:

1. **How does the inclusion of different audio features impact the results?**

   As the conditioned input for the proposed model, suitable features extracted from audio are crucial for generating desired motion. Understanding how different representations affect the results allows us to identify what audio features are the most informative and easiest to relate to motion.

2. **What representative features of interactive dance motion are most suitable for quantitative evaluation?**

   Features typically used to measure the quality of generated dance movements have recently been shown to be at odds with human evaluations [5]. This suggests the need to determine what features are most representative of the movement.

3. **How does the amount of data used to train our model affect the resulting motion?**

   Diffusion models typically benefit from large amounts of training data. Given the lack of suitable datasets available in this domain, we experiment with varying levels of data scarcity and artificial augmentation to determine the impact on our model.

1.2 Thesis Outline

We first look at the most relevant literature in the realm of human motion synthesis including motion representations, common synthesis tasks, and audio-driven generation. Then we detail our approach to tackling the problem of audio-driven partner dancing animation. We reason as to what representations of audio and motion we will use, as well as why our proposed architecture was chosen. We detail our experimental setup and the metrics we choose to evaluate against our results.

2 Relevant Literature

Starting with understanding how current methods represent the human skeleton, we compare varying representations of pose. Next, the most common tasks associated with motion synthesis are introduced where works are separated by their underlying architecture. We discuss the difference between motion prediction and generative synthesis before moving on to music-driven dance generation. Here we review the most significant contributions once again separated by their architectures. Different audio representations and quality measures are reviewed. Then, we briefly touch on work associated with human-human interaction generation and compare reactive methods to interactive methods before moving to relevant dance datasets.

2.1 Motion Synthesis Representations

3D rigged skeletons are typically used for animating human characters as it is more trivial than animating the individual polygons of a mesh [21]. The skeleton consists of segments connected by joints that have several Degrees of Freedom (DOFs). The orientation of the joints defines the character's angular pose representation, while the coordinates of the joint positions define the positional pose representation [22]. Human motion can therefore be represented as a sequence of poses separated by some time step.

2.1.1 Positional Pose

Joint positions are a convenient and intuitive way to represent pose. As joint positions are most often embedded in the Cartesian coordinate system, they can easily be interpolated and visualized. However, joint positions do not inherently constrain bone lengths to remain constant. This leads to a reliance on additional constraints or the need for joint reprojection onto a valid configuration [23–25]. They are considered to be an incomplete representation for modeling a 3D character as they lack bone orientation information and can lead to ambiguous recovered joint rotations when using inverse kinematics as shown in Figure 1 [26].

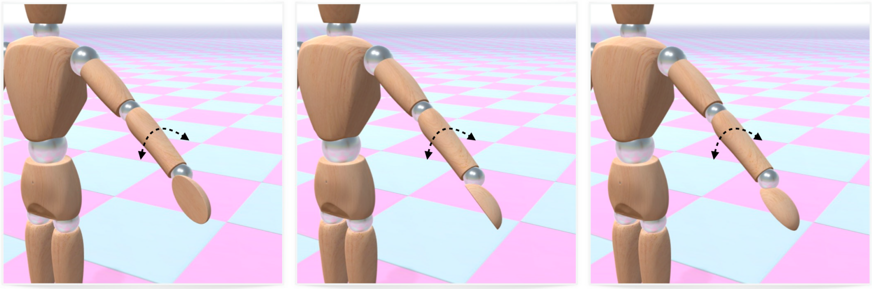


Figure 1: Varying limb rotations can connect the same set of fixed 3D joint positions. Thus, joint positions alone are not sufficient for modeling natural motion [26].

2.1.2 Angular Pose

Angular or rotational representations avoid issues relating to bone stretching by allowing for a parameterized skeleton [27]. In this way, the skeleton can be seen as a hierarchical set of joints where each joint is described with respect to its parent using the joint rotation and the bone dimensions [22]. The most intuitive representation for describing joint rotations is *Euler angles.* They represent 3D orientation as three successive rotations relative to a fixed coordinate system *x, y, z.* When two axes are aligned, one DOF is lost resulting in a gimbal lock. In addition, Euler angles are discontinuous and multiple conventions exist for defining the order of the axes. While there are ways to overcome the aforementioned issues, they often result in being less computationally efficient than the alternatives [23]. *Exponential map* representations have been utilised for motion synthesis [28] and motion prediction [29], but suffer from singularities resulting in losing a DOF. *Quaternions* have been used frequently to represent joint rotations since Pavllo et. al introduced a quaternion-based motion prediction framework called *QuarterNet* [23, 30]. As quaternions lie in $\mathbb{R}^4$ it was thought they would avoid the discontinuities and singularities of Euler angles and exponential maps which both lie in $\mathbb{R}^3$. However, it was shown by Zhou et. al that all representations in $\mathbb{R}^n$ with $n \leq 4$ are discontinuous and that neural networks can learn better from continuous representations [31]. Instead, they proposed a continuous 6D parameterization which has seen use in recent works, including VAE [32] and diffusion-based [33–36] motion synthesis methods.

2.1.3 Hybrid Approaches

Several works have seen success by combining positional and angular pose representations. Aberman et. al [37] divide a motion sequence into a static component consisting of 3D positional offsets and a dynamic component consisting of quaternions describing the rotation of each joint. Their motion retargeting framework uses separate branches for the dynamic and static components. This leads to a reduction in error over non-skeleton-aware approaches where there is no division. Shi et. al [26] chose to divide motion representation into static bone lengths and dynamic joint orientations (quaternions) to then use forward kinematics to recover the joint positions. In addition to joint angles and positions [38], others have incorporated joint velocities [39, 40], and even linear and angular velocities [41, 42].
Andreou et. al [43] recently proposed a dual-quaternion representation for pose where both positional and rotational information is encoded using a joint current configuration (i.e. relative to the root). They note that joint local configurations – joint information w.r.t the parent joint – are vulnerable to accumulation errors. They found their representation generates more stable motion than both the 6D and single quaternion parameterizations when using recurrent architectures.

2.2 Motion Synthesis Tasks

When looking at deep learning methods for motion synthesis, multiple challenges arise. Models must account for the fact that motion is aperiodic. They must also be able to capture spatial and temporal dependencies, such as the dependency between the right arm and left leg during a walk cycle. The produced motion should be plausible based on past observations. Finally, humans can exhibit complex behaviors that are dependent on their environment and are rarely acting in complete isolation. Deep learning models should therefore take other humans or objects into consideration when predicting motion [22]. This can be done through conditional generation, where additional information is supplied as an input signal to guide the learning. We will first look at some of the seminal works in motion prediction separated by technique as these form the foundation on which other tasks build. We then discuss the task of generative synthesis and compare its goal with that of prediction.

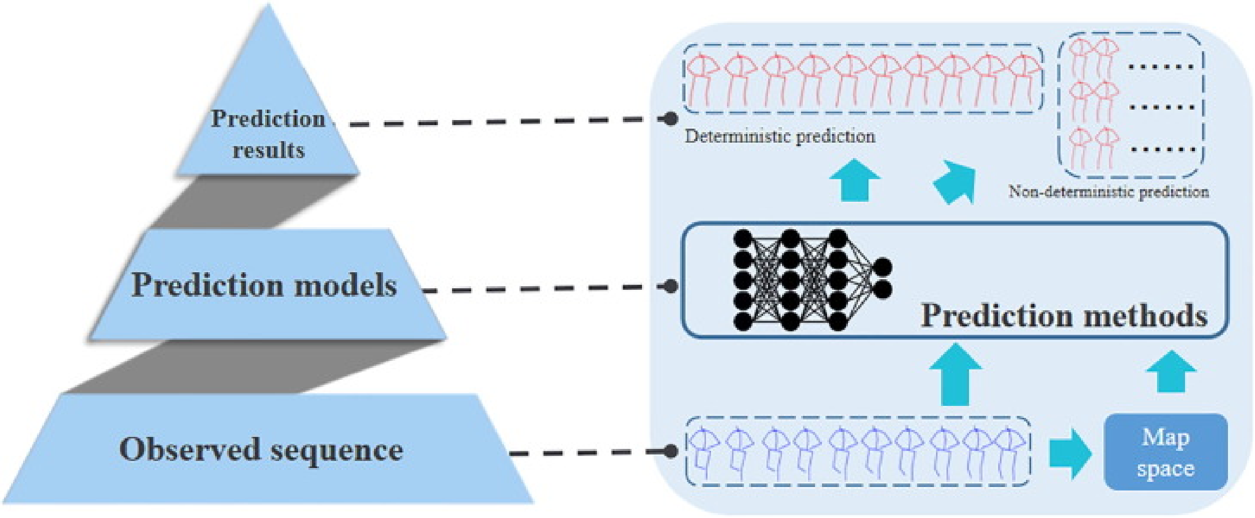


Figure 2: Illustration of the motion prediction task. Illustration from Lyu et al. [44].

2.2.1 Prediction

Human motion prediction forecasts the future motion of an individual based on their previous motion. These models rely on the input motion data's temporal and spatial structure, learning from past frames to generate probable future motion (see Figure 2). Statistical models for this problem include the use of Markov Chains [45], Dynamic Bayesian Networks [46], and Gaussian processes [47]. For the purpose of this proposal, we will focus on common deep-learning methods used to tackle this task. These include the following frameworks: Recurrent Neural Networks (RNNs), Graph Convolutional Networks (GCNs), and Generative Adversarial Networks (GANs).

RNNs RNNs are often used for sequence-to-sequence (Seq2Seq) tasks. In a Seq2Seq architecture, an "encoder" network receives an input sequence and generates an internal representation. A second "decoder" network then takes the internal state and decodes the target sequence from the generated internal representation [48]. Fragkiadaki et al. [49] proposed the Encoder-Recurrent-Decoder (ERD) architecture where an LSTM (long short-term memory) network is trained in the hidden layer of an autoencoder (see Figure 3). They found the network was prone to accumulation errors and would produce implausible results so they gradually added random noise to the input during training. Martinez et. al [29] trained their model in an auto-regressive fashion by feeding the network predictions instead of random noise to better recover from its own mistakes. They also opted to use a GRU (Gated Recurrent Unit) module instead of an LSTM network. While this method produced suitable results for short-term prediction (<500ms), they showed it's difficult to also generate long-term (>500ms) forecasting using RNNs. Generation of long-term predictions can suffer from mode collapse where predictions converge to a mean pose and diversity is lost. When trained on a single action combined with a sampling-based loss, long-term forecasting was plausible but short-term predictions suffered from discontinuities. Training their model with residual connections on multiple actions resulted in better short-term predictions, but converged to a mean pose. As the hidden layer of the previous unit's output is used for the prediction of the next unit, errors can easily accumulate. This accumulation leads to discontinuities and unrealistic output, particularly in the long term. Gui et al. [50] incorporated adversarial training into their GRU-based Seq2Seq architecture to improve the long-term forecasting results. Inspired by GANs, they introduce two sequence-level discriminators to ensure predicted motion is human-like and coherent with the input sequence. Despite seeing improvements, the produced motion is still limited to "brief" long-term forecasting (<4s).

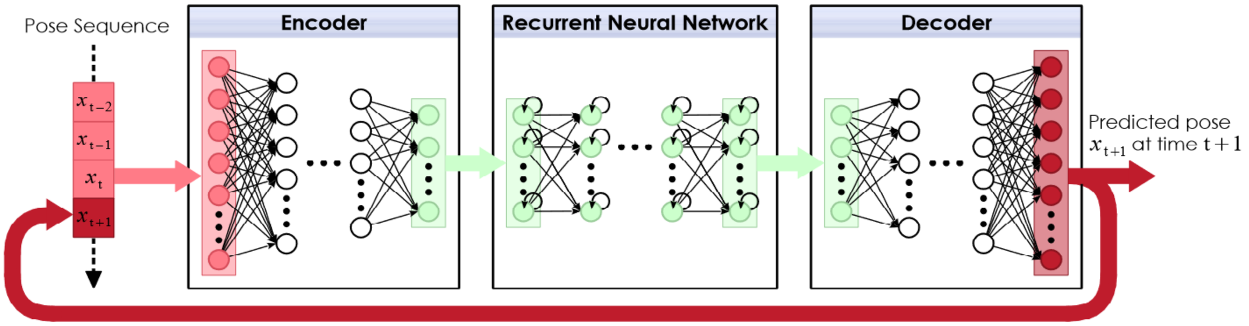


Figure 3: Illustration of the Encoder-Recurrent-Decoder (ERD) archiecture proposed by Fragkiadaki et al. [49]. Illustration from Mourot et al. [22].

GCNs Graph Convolutional Networks can take advantage of the graph nature of the human skeleton. The first usage of this architecture for motion prediction was by Mao et. al [51] where human motion sequences are represented through discrete cosine transform (DCT) coefficients. This representation is then fed into a GCN to learn the temporal and spatial inter-joint dependencies. Later, they proposed an attention-based feed-forward network that leverages the idea that human motion often repeats itself [52]. Their motion attention model looks at sub-sequences of past motion and compares them to the current observation. The future sequence is then computed as a weighted sum of sub-sequences. [53–55] use multi-scale graphs to capture the skeleton structure from individual joints to increasingly large bodily structures. This type of model can more easily represent the internal relations of the body. Because a group of joints will often come from the same body part, gradually abstracting the skeleton can greatly improve the stability of the predicted motion [53]. These models often fall short of learning the temporal dependencies of the motion, especially when translated into the frequency domain.

GANs One major drawback to some GCN and RNN-based methods is the issue of error drift. As the predicted pose is ultimately used as input for future poses, the prediction error will propagate into the next pose. This issue is less present with GANs as multiple poses are predicted at once. A generator takes in random noise as input to generate a motion sequence. The discriminator takes in both the real and generated data and attempts to distinguish the two. GANs were first introduced to the task of human motion prediction by Barsoum et. al [56] with HP-GAN. HP-GAN learns a probability density function of future human poses conditioned on the previous pose. This approach doesn't suffer from the aforementioned error drift issue. However, stable motion isn't guaranteed as there is no way to ensure training convergence. In addition, their results weren't evaluated against any state-of-the-art (SOTA) deterministic methods. Kundu et al. [57] introduced the Bidirectional Human motion prediction GAN, or BiHMP-GAN which regresses a randomly sampled extrinsic vector. This supports the learning of forward and backward mappings between the stochastic input and the generated motion, reducing the chance of mode-collapse. BiHMP-GAN outperformed deterministic SOTA methods and HP-GAN in both short and long-term predictions. Cui et al. [58] incorporated Temporal Convolutional Networks (TCNs) with GANs in order to use a hierarchical convolution structure to model long-term predictions. This structure has been shown to be more efficient than RNNs in computational complexity, the number of model parameters, and parallelism. Spectral normalization is embedded directly in the model to avoid mode-collapse. Liu et al. [59] presented AM-GAN whose structure consists of local GANs for different body parts which are aggregated via a global GAN. This divide-and-conquer approach to modeling different body parts and then merging them at the end proves to outperform both HP-GAN and BiHMP-GAN.

Three popular frameworks for human motion prediction were presented: RNNs, GCNs, and GANs. While RNNs succeed in modeling temporal dependencies in the motion data for short-term prediction, long-term forecasting often results in mode-collapse. GCNs use the graph-like structure of the human skeleton to model spatial dependencies and have been shown to outperform RNNs, but they fail to adequately capture the temporal dependencies. Both RNNs and GCNs suffer from error accumulation when a singular pose is predicted from the prior. GANs resolve this issue by predicting motion sequences instead. They are also capable of generating diverse sets of motion which is useful for predicting complex and variable motion. However, GANs are more difficult to train, and balancing the generator and discriminator requires careful consideration.

2.2.2 Generative Synthesis

In contrast to motion prediction, generative synthesis produces novel human motion representing the distribution of the underlying data without the use of an input sequence. Instead of prior motion frames, deep learning frameworks can use random seeds to generate samples from a learned latent space. Motion prediction and generation both utilize deep learning for the modeling of spatial and temporal dependencies. While prediction seeks to produce the most probable continuation of motion, generative synthesis aims to produce diverse motion sequences and is often conditioned on some external input signal. Generative approaches can be divided into three categories: VAE-based, GAN-based, and diffusion-based.

VAE-Based Variational Autoencoders (VAEs) consist of an encoder followed by a decoder. While there are similarities to the structure of the ERD architecture used in motion prediction, the components differ in their purpose. The VAE encoder maps the input to a lower-dimensional latent space representation modeled as a multivariate Gaussian distribution. The decoder maps the latent space representation back to the input. Once the network is trained, the VAE samples random values from the learned latent distribution which are then fed into the decoder for the generation of data. These results in a non-deterministic output for a given input sequence unlike with the ERD. Habibie et al. [60] combine VAE samples with control signals through an RNN to synthesize predicted and novel motion. Guo et. al [61] proposed Action2Motion which uses a conditional VAE to handle pose generation per time step while an RNN models the temporal dependence. An action label is provided to the encoder along with previous pose states. Instead of generating individual poses, Petrovich et. al [62] introduced an Action-Conditioned TransfORmer VAE (ACTOR) which is capable of synthesizing variable-length motion sequences. They provide an action label as input to both the encoder

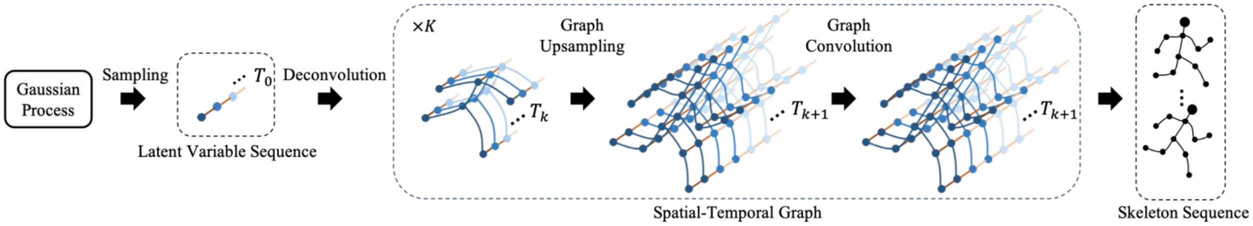


Figure 4: Illustration of the Convolutional Sequence Generation Network (CGSN) proposed by Yan et al. The CSGN gradually upsamples the spatial-temporal graphs before finally transforming the latent signals into a sequence of human skeletons [64].

and decoder and see improvements over Action2Motion.

GAN-Based Unlike prediction, these GAN-based methods don't utilize the past motion sequence as input. They instead rely on noise and may incorporate an input signal such as an action label. Cai et al. [63] propose a two-stage network in which a conditional WGAN first generates a single frame to model the spatial component of the pose. A conditional sequence GAN finds the optimal latent vector sequence which looks natural to model the temporal component. The second stage deals with generating a video based on the synthesized human pose sequence. Yan et al. [64] instead jointly model the spatial-temporal components using spatial-temporal graph convolutions. Both the generator and the discriminator are formulated as GCNs where graph pyramids are used to gradually upscale/downscale the latent vector sequence (see Figure 4). The latent vector sequence is sampled from a Gaussian process prior. Degardin et al. [65] note that CSGN's reliance on Gaussian processes limits the variability between samples. They instead propose Kinetic-GAN which generates motion from a single latent point. The inclusion of an intermediate latent space results in easier generation of realistic actions and outperforms CSGN.

Diffusion-Based Diffusion models [66] have gained significant popularity for generative synthesis in the domain of image generation [67, 68]. Denoising Diffusion Probabilistic Models (DDPMs) gradually add noise to an original data sample and then learn the reverse process of gradually denoising the sample. Once trained, the model can generate a new sample by denoising random noise. As they were shown to outperform GANs for the task of image synthesis [69], researchers naturally looked into their potential for motion generation. Kim et al. [33] and Zhang et al. [35] both proposed text-to-motion diffusion-based models one day apart. The datasets these models were trained on contain motion clips paired with sentence descriptors such as "A person is waving their left hand." MotionDiffuse [35] outperformed Action2Motion as well as ACTOR in action-conditioned motion generation. Most generative models are regularized using geometric losses to enforce physical plausibility such as joint, velocity, and foot contact consistency. However, this is non-trivial in the case of diffusion as it's the noise that is predicted. Tevet et al. [70] introduced the lightweight Motion Diffusion Model (MDM) which predicts the sample rather than the noise at each diffusion step. This leads to the enabling of geometric losses to improve results. Zhang et al. [71] employ retrieval techniques to enhance the generalizability of their model. They show improved text-to-motion performance over MotionDiffuse and MDM especially in the case of uncommon motions.

Diffusion models have continuously been shown to outperform both VAEs and GANs in the task of generative synthesis. They are capable of producing more diverse motions at the cost of longer training times.

2.3 Music-Driven Dance Generation

Human motion and music are strongly intertwined. When listening to a song, humans tap their feet, wave their hands to the beat of the music, sway with their bodies, and can produce complex dance choreographies. These more complex motions can be produced as an artistic expression of an idea, as a way to pay homage to cultural heritage, or simply for the dancer's own enjoyment. Given this strong correlation, research on generating human motion from audio has been an active topic of research for decades and is only growing in popularity with the improvements in deep learning techniques. This research has a multitude of applications including cultural heritage preservation [72], instructional training [73], as well as improving the realism of virtual characters in digital experiences. In the following sections, we distinguish between single person and multi-person motion generation.

2.3.1 Single Person Motion

The first work to propose rhythmic motion generation from audio was Kim et al. [1] by using motion transition graphs. Their scheme requires music to have similar rhythmic patterns to the desired motion and is limited in the variety of music which can be used. Shiratori et al. [9] noted that the intensity of a dancer's movements depends on the associated intensity of the music. Using rhythm and intensity features of the music, they synthesize new dance motions by extracting motion segments from a motion capture database. Sauer et al. [10] improved upon previous works by using dynamics and rhythm to affect the magnitude and length, respectively, of primitive motions. More recent works have taken advantage of the ever-growing popularity of deep learning techniques.

LSTM-Based Crnkovic-Friis et al. [74] first employed LSTMs to the problem of dance synthesis with Chor-RNN. Their approach is capable of producing novel choreographic movements, however, Chor-RNN isn't conditioned on any audio input. Alemi et al. [11] found that Factored Conditional Restricted Boltzmann Machines were easier to train than LSTMs on smaller datasets and also produced quicker results. Their proposed architecture GrooveNet is capable of learning the cross-modal mapping of music information and motion sequences in order to produce dance movements from new audio. Given the small dataset used, GrooveNet highly overfits and is incapable of generalizing beyond the songs used in training. A lack of training data led Tang et al. [12] to release an audio-to-dance paired dataset containing four types of dance. They also proposed an LSTM-autoencoder approach for learning a mapping between acoustic and motion features. Qi et al. [13] utilised a Seq2Seq architecture combining LSTMs with Self-Attention mechanisms which proved to outperform an LSTM baseline. The primary drawback of LSTMs for dance generation is that given the same audio, the resulting motion will be the same. This approach thus lacks diversity which is paramount for dancing given individual styles.

Transformer-Based Huang et al. [16] utilize a transformer-based music encoder followed by an LSTM-based decoder to generate 2D dance sequences with DanceRevolution. The Two-Stream Motion Transformer (TSMT) model was proposed by Li et al. [4] where movement and music are processed separately and then fused later to predict 3D pose at the next time-step. Li et al. [3] point out that TSMT generates unrealistic motion due to the late fusing. They proposed $FACT$ (full-attention based cross-modal transformer) which better preserves the correlation between music and motion and produces more realistic long motion than DanceRevolution and TSMT. DanceFormer [2] outperforms both TSMT and $FACT$ by generating key poses and then predicting the motion curve between the poses based on the audio. These methods are used for forecasting and fail to adequately learn the correlation between audio and motion.

GAN-Based Lee et al. [14] teach their model "how to move" by decomposing a dance into a series of dance units. These units are short snippets of fixed-length poses which are then modeled by a VAE. A cross-modal GAN is utilised to generate a sequence of movements conditioned on the input music. At runtime, style and beat information is extracted from the music and a 2D sequence of dance units is generated and warped to match the beat. This approach runs the risk of warping a unit to an unrealistic speed depending on the speed of the music. Sun et al. [15] avoid the post-processing step of aligning the movement to the music when predicting a 3D dance sequence from input music. These methods suffer from cumulative errors when generating very long sequences leading to drift of the dancer's root joint.

Diffusion-Based The first use of diffusion for generating dance was by Alexanderson et al. [6] by adapting the DiffWave architecture [75] to model 3D pose sequences. Instead of dilated convolutions, the authors utilize Conformers which outperform both Transformers and CNNs on tasks like speech recognition. No comparison between their approach and SOTA methods was made. Tseng et al. [5] introduced the transformer-based diffusion model EDGE (Editable Dance GEneration). Able to generate motion sequences of arbitrary lengths, EDGE outperformed previous SOTA works both quantitatively and qualitatively with its inclusion of Jukebox [76] for extracting audio features. In addition, users can provide temporal and joint-wise constraints on the generated motion to allow for editing and motion in-betweening. While promising and more stable to train than GANs, these models require large amounts of data and have long training times.

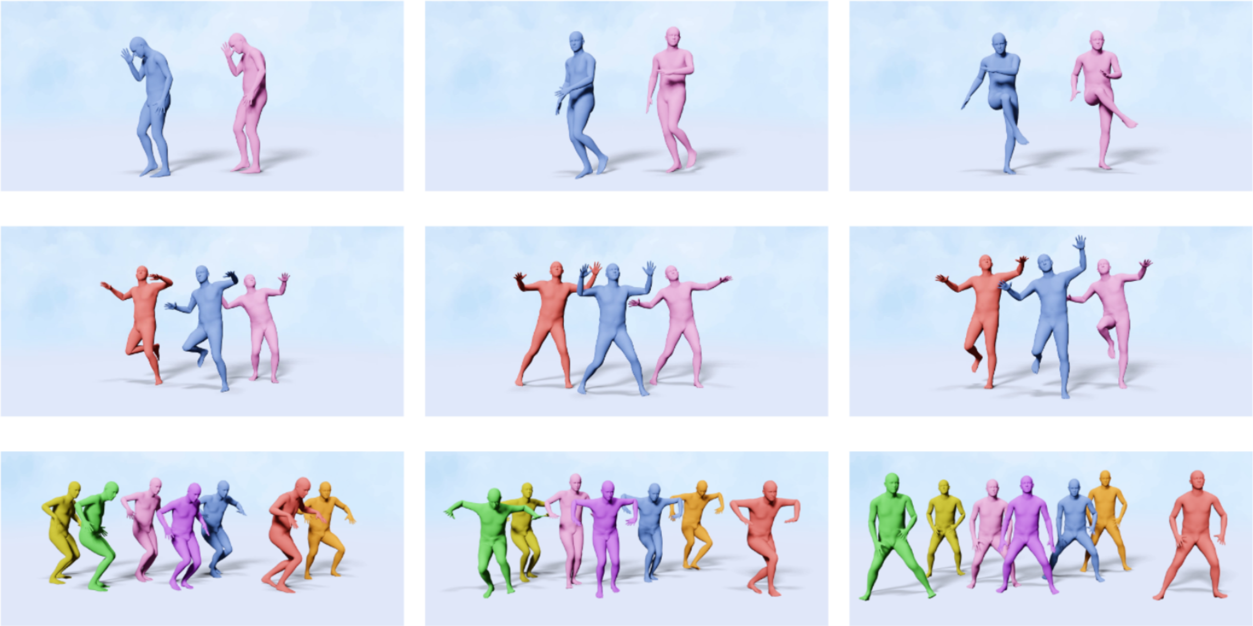


Figure 5: Examples of generated motion from GDanceR. Note that all dancers are performing a similar choreography and there is no physical interaction between dancers. Figure from [17].

2.3.2 Multi-Person Motion

As of the writing of this proposal, there exist only two papers to synthesize motion for multiple dancers conditioned on audio. Wang et al. [18] first proposed GroupDancer to attack the problem of audio-driven multi-person dance synthesis. Their approach first decides when and which dancers are collaborating at any given time, then predicts each dancer's motion, and lastly generates fluid motion sequences. The author's idea of collaboration is when two or more dancers are dancing at the same time. This is different than that of partner dancing where one dancer's movement influences the other. There is no spatial awareness between dancers and it is unclear how the initial positions of the dancers are determined.

Le et al. [17] propose a Music-driven 3D Group Dance generator (GDanceR) that takes in a music sequence and initial dancer positions to generate coherent dance movements attuned to the music. To generate group motion, an RNN captures the temporal dynamics of each dancer and cross-attention mechanisms encode the spatial relationship of all dancers. As people with closers positions tend to have higher motion correlation, they use the distance between dancers as a bias in the attention weight. This encourages awareness and interactivity to be higher when dancers are nearer. They modify *FACT* to generate motion for multiple dancers and compare their results against this modification. It is not clear why GDanceR does not compare its results to that of GroupDancer. GDanceR produces superior performance as *FACT* produces bodily intersections given its complete lack of spatial encoding between multiple dancers (see Figure 5).

2.3.3 Audio Features

The representation of the input is a crucial factor for generative synthesis as models need to learn a mapping from audio features to a motion sequence. Dancers may align their movements to that of a specific instrument, while others will convey their emotional response based on individual interpretation. It's important for the representation to be indicative of what humans would hear when moving to the music. Kim et al. [1] first used a MIDI (Musical Instrument Digital Interface) representation in order to easily provide the music beats directly as MIDI allows for features such as onset, pitch, and volume to be easily obtained [9]. Shiratori et al. [9] utilize Constant Q-Transform (CQT) to extract the spectral energy of each note in their audio input. Using CQT feature vectors they find repeating patterns defining the structure of the music. Rhythm and intensity are also extracted from these CQT feature vectors to affect the duration and impact of the motion. With advances in music information retrieval (MIR), GrooveNet [11] utilised a large set of audio features [78]

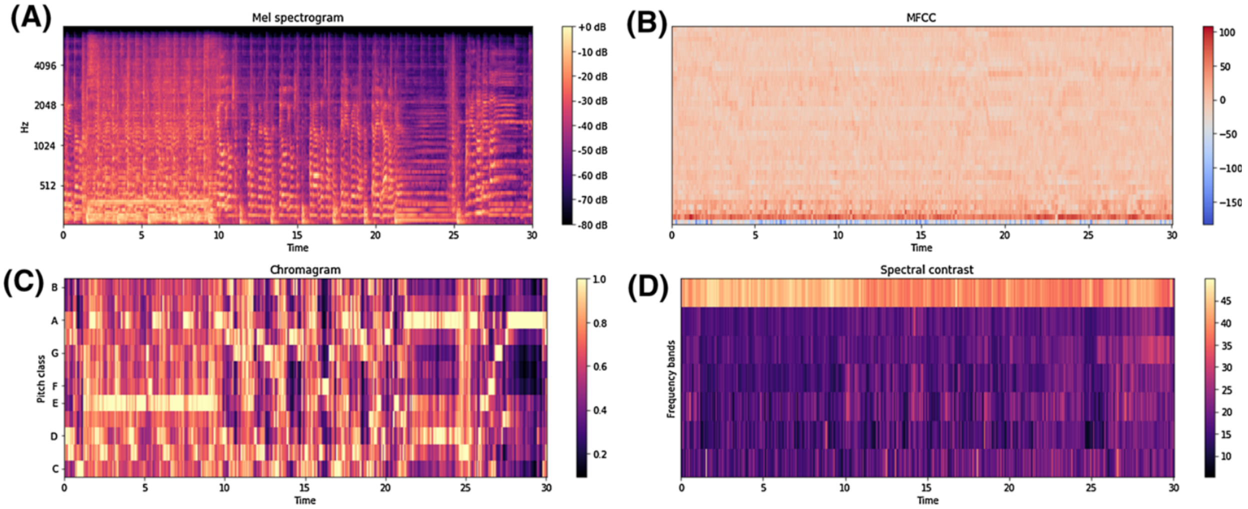


Figure 6: Visualization of musical features. A shows the mel spectrogram. B shows the first 40 MFCCs. C shows the chromagram. D shows the spectral contrast. Figure from [77].

including low-level features, spectral features, timbral features, and melodic features. One of the most notable features in this set are the Mel-Frequency Cepstral Coefficients (MFCCs). MFCCs attempt to mimic the human auditory system which is more sensitive at lower frequencies and less so at higher frequencies. They are often used in speech-driven facial animation [79, 80] as well as music-driven gesture animation [81]. Tang et al. [12] combined MFCCs with a constant-Q chromagram, tempogram, and onset strength whereas Qi et al. [13] only used MFCCs. *Bailando* [82], GDanceR [17], and *FACT* [3] use the same features as [12] with the inclusion of MFCC delta. In all aforementioned approaches, no experiments are performed to see the impact different feature representations have on the results. Li et al. [4] showed the effect of using only time intervals between beats, only MFCCs and their delta, or a combination of both. While using only beat information resulted in higher diversity scores, the combination showed to produce the highest quality motion. Dabral et al. [8] represent music signals through the Mel spectrogram for MoFusion. The Mel spectrogram differs from MFCCs in that it uses a linear spaced frequency scale while MFCCs uses a quasi-logarithmic scale. There is little discussion as to why authors use one audio representation over another which leaves room for future investigation. See Figure 6 for a visual representation of some of the discussed features.

Recent advancements in audio-synthesis led Tseng et al. [5] to instead use the audio representations learned by the generative model Jukebox [76]. They note that the music representation is equally as important as the generative model used. Jukebox representations were shown to yield stronger performance on a series of MIR tasks over other pre-trained models [83]. The authors compare the performance of EDGE with Jukebox against EDGE with the same features as Li et al. [3]. They showed a significant user preference for the model with Jukebox representations.

2.3.4 Evaluation

Evaluating the results of generated dance motion is an ever-evolving task. We can divide the evaluation into quantitative metrics and qualitative perception studies. We will touch on the most common statistical metrics and what features users are asked to consider when evaluating these dances.

Quantitative The most common method for evaluating dance quality uses the Fréchet Inception Distance (FID) [84] to measure the distribution distance between generated motion and the ground truth. Initially, works would contrastively train a classifier on the collected data as a type of feature extractor to generate the FID [14, 16]. Li et al. [3] instead opted to use well-known feature extractors for motion-capture data [85, 86]. The features are divided into kinetic and geometric resulting in a calculation of FID_k and FID_g respectively. Despite these measures becoming a standard for motion quality, Tseng et al. [5] argue that they are unreliable. In their experiments, they showed that the ground truth had a much higher FID_g score than automated methods. They also saw that *FACT* had the best FID_g score but performed the worst in user studies. EDGE also produced higher quality results with longer training, but FID$_k$ worsened in the latter half of the training. They believe that FID$_k$ and FID$_g$ only compute superficial features of the data and that more representative features should be used to make these metrics reliable.

Generative models should be capable of producing a diverse set of motions for different songs. This generation diversity is normally calculated using the average feature distance in the same kinetic and geometric feature spaces used when computing the FID. Tseng et al. [5] still use these features to compute their diversity score despite their critiques. They claim that diversity shouldn't be maximized but should instead be close to the ground truth as jittery or unnatural movements can result in higher scores. Some more general human motion works decided to use the learned motion features from a contrastively trained classifier as the basis for computing diversity [61]. Connected to diversity is the multi-modality metric which measures the distance between varying output motions for the same input. This metric is crucial for evaluating generative dance models as people will interpret music differently and that should be reflected in the movement.

An important aspect of any partner dance is the ability of the dancers to follow the beat of the music. Both Siyao [82] and Li [3] propose a Beat-Align Score (BAS) which indicates how close the kinematic beats (local minima of joint speed) line up with musical beats. Tseng et al. [5] once again critique this metric as musical beats are a "loose guide" for the rhythm and timing of the motion. This claim applies to specific genres of dance where the movement is less structured. For some partner dances (e.g. Salsa, Bachata, Waltz) the dancers are expected to make their first step on a specific beat of the music. For example, when dancing "On-1" Salsa where the music is normally subdivided into 8 counts, the leader and follower should both make their step on the first count or the "1". Aristidou et al. [7] compute the ratio of kinematic beats to musical beats as well as the ratio of aligned kinematic beats to the total kinematic beats. They observe the ratios in the ground truth aren't one-to-one and aim to achieve a similar ratio with the generated motion.

Few metrics have been introduced to tackle the problem of multi-person dance generation. Le at al. [17] introduce three group metrics: Group Motion Realism (GMR), Group Motion Correlation (GMC), and Trajectory Intersection Frequency (TIF). GMR measures the degree of realism between the generated and ground-truth group dance motions by finding a single unified representation for all dancers' motions based on kinetic features. The idea of using a single metric to represent all dancers' motions is promising. However, as this is a recent addition it remains to be seen how representative GMR is of multi-person motion. GMC measures the synchronicity between the movements of the dancers by aligning their motion sequences and calculating the mean cross-correlation using kinetic features. This metric is not well suited for partner dancing where the motions of dancers are not expected to be aligned, e.g. when the follower turns and the leader is completing basic steps. TIF measures how frequently the dancers' meshes intersect and is best applicable for choreographies where no physical contact is expected.

Qualitative Tang et al. [12] note different dancers will use unique and personal movements for the same song. Therefore, relying solely on the error between the output and ground truth is meaningless. They proposed that user evaluations are a necessity when judging the results of dance generation. They asked users to rank the performance of variations of their model. Then they showed the generated dance on two different songs: the input audio and another song of a different genre but with the same tempo. The users were asked which song better fit the motion. Neither study provided any indication of physical realism nor preference for generated motion against the ground truth.

Li et al. [3] first gauged their participants' dancing experience and asked how often they watch dance videos. The participants were then presented with a side-by-side comparison of *FACT* against baseline methods and ground truth and asked "*which person is dancing more to the music? LEFT or RIGHT*". Siyao et al. [82] ask a similar question for *Bailando*. Both of these questions only determine the style consistency of the motion with music. Huang et al. [16] asked participants to select preferred dances based on: motion realism, style consistency, and smoothness independent of music. These questions better reflect the overall quality of motion in contrast to those of *FACT* and *Bailando*. Aristidou et al. [7] ask participants to rank realism, style consistency, and beat-to-motion synchronization. Their survey demonstrates a clear difference in ratings depending on the user's experience with dance.

In order to compare more than two models, Tseng et al. [5] utilize the chess-based Elo ranking to capture the relative quality of EDGE against *FACT*, *Bailando*, and the ground truth. Users were also asked to select the dance that "looked and felt better overall". This phrasing is an improvement upon the singular style consistency question, however, it doesn't allow room for interpretation of *why* users prefer one motion over another.

2.4 Human-Human Interaction Generation

While single person motion generation has been an active research area for decades, generating motion for multiple interacting bodies is relatively unexplored. This proves to be a difficult task as models must learn body-to-body motion correlations as well as inter-body joint relations. Works in this field can be divided into reactive and interactive. Reactive motion correlates to motion that is generated in response to another individuals motion. Interactive motion generation is applicable when multiple people are acting synergistically and must continuously take each other's motion into consideration.

2.4.1 Reactive Motion Generation

Reactive motion generation can be defined as generating the motion sequence of Person A given that of Person B. Baruah et al. [87] note that perception and action are strongly correlated. For example, when boxing, each person pays close attention to the position and movement of the arms in order to block or evade a punch. They introduce an attention-based agent which learns to attend to the most informative joints of both the other person as well as itself. The authors use a Multimodal Variational RNN (MVRNN) where the next frame of reaction is predicted using previous motion frames in conjunction with perceptual and proprioceptive information. Chopin et al. [88] were the first to utilize an attention-based Transformer framework for human reaction generation. They improved their model's spatial attention using a skeleton adjacency graph for self-attention. By introducing the distance between the joints of both skeletons during attention, they saw further improvements. While able to produce higher quality dance movements than Baruah et al. [87], the generated motion lacks diversity as it is a deterministic approach. As previously discussed, diversity is a fundamental component of dance generation and so deterministic approaches are not suitable for our goal.

Attention mechanisms seem fitting for generating realistic partner dances. When dancing salsa, a leader can indicate they want the follower to turn by raising one hand and pushing slightly. In this scenario, the follower's attention is predominantly on the hand of the leader and not on the rest of the body. A framework that incorporates the more salient areas of another person is likely to produce better results. While this leader-follower dynamic seems to fit into the task of reactive motion, the leader also relies on the follower's positional information to determine what move to lead next. Realistic dance and other complex inter-person interactions require all persons to attend to each other.

2.4.2 Interactive Motion Generation

Kundu et al. [89] introduce a cross-conditional RNN (ccRNN) framework for synthesizing long sequences. They use separate ccRNNs for each dancer to simulate the leader-follower dynamic. In this way, they alternatively generate the motion of each person based on previous generations. This approach is unfortunately prone to error drift and longer generated motion shows a decrease in quality. Guo et al [90] propose a cross-interaction attention (XIA) mechanism capable of predicting cross dependencies between two pose sequences. Two pipelines, one for the leader and one for the follower, use an attention model for learning temporal attention and a GCN predictor to model the spatial attention among joints. The XIA modules share information between the two motion predictors before learning temporal and spatial attention. This sharing of information between predictors outperforms single-motion prediction models adapted for multiple persons. Rahman et al. [91] demonstrate the effectiveness of GCNs and DCT motion representations for two-person forecasting. They showed that the inclusion of attention via a Graph Attention Network in their model did not result in any performance gain and substantially increased the number of parameters. They outperform XIA and show that adopting a multi-scale hierarchical motion representation is a limiting factor for achieving the best performance.

All aforementioned methods deal with motion prediction, where the input to the network is the ground truth. Very few works have tackled the generative synthesis of human-human interaction. Shafir et al. [20] expand MDM [70] to simultaneously generate interactive motion. By employing a fixed generator for each person, they introduce a single-layer transformer model ComMDM which coordinates between the two MDMs. This method is capable of synchronizing motion-from-text between two priors, however, it is limited to interactions seen in training and lacks generalizability. Liang et al. [19] propose two cooperative transformer-style weights-sharing networks with mutual attention (see Figure 7). Using a novel representation of human motion, they

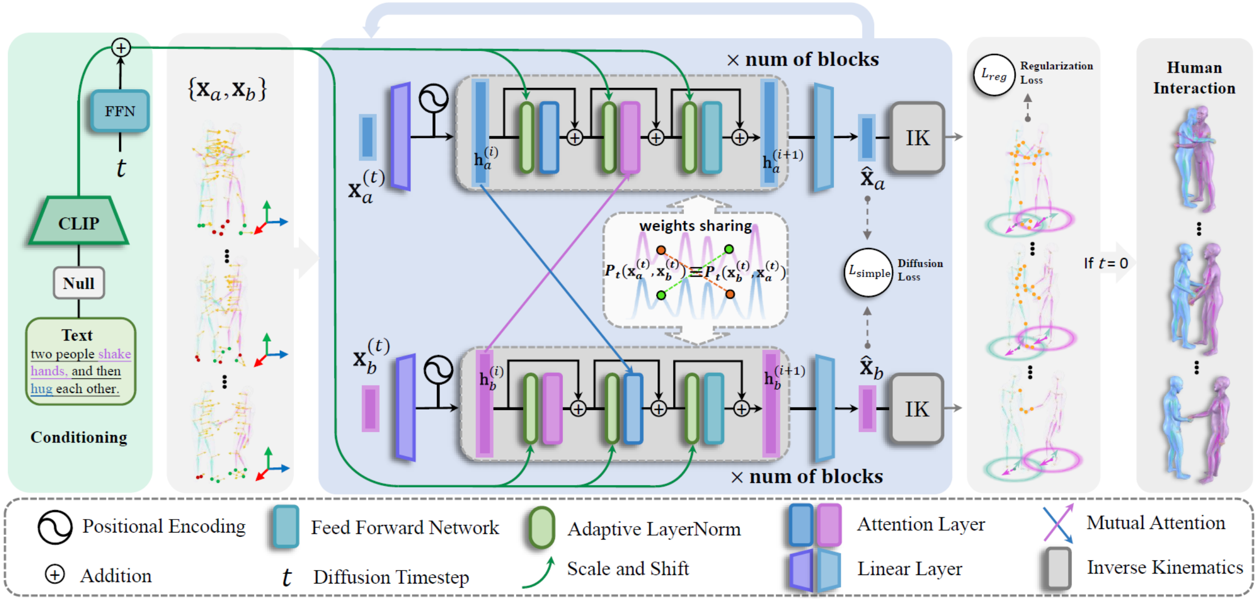


Figure 7: Illustration of the InterGen architecture proposed by Liang et al. [19].

demonstrate using global joint information avoids drifting caused by accumulating local angular and linear velocities. Their framework also allows for generative reactive motion of one person using the other's motion as input.

2.5 Relevant Human Motion Datasets

To the best of our knowledge, there are no publicly available datasets consisting of partner-dancing motion paired with audio. Kundu et al. [89] published the *DuetDance* dataset with their ccRNN technique. This dataset consists of motion curated from dance tutorial YouTube videos for a large number of partner dances including Salsa, ChaCha, Jive, Rumba, and Samba. Liang et al. [19] contributed *InterHuman*: a multimodal dataset with over 107M frames of interactive motion. While most of the motion is daily motion and combat, there are sequences of Latin, Waltz, and other dancing styles. Despite claims of publishing *DuetDance* is not available. The Carnegie Mellon University Motion Capture Database (CMU MoCap) [92] features a short collection Salsa dancing for a leader and follower. None of the listed datasets are paired with audio.

3 Methodology/Approach

Our goal is to generate realistic partner dancing motion conditioned on a piece of music. The produced motion should be representative of the dance genre associated with the input music. This requires a generative model capable of learning mappings from audio features to motion features. To this end, we propose DuoDance: an audio-to-motion diffusion model capable of generating realistic partner dances conditioned on music. We update the architecture of InterGen to accept audio as conditioning input. To ensure the conditioning has a strong influence on the generated motion, we introduce a novel attention-based block consisting of self-attention, cross-attention, and mutual-attention. We first discuss our choice of motion representation for both the input and output of DuoDance. We then discuss our choice to compare varying audio representations as conditioning input. Finally, we present the framework for DuoDance, including diffusion architecture, conditional encoding, and loss choices.

3.1 Motion Representation

In HumanML3D [32], Guo et al. propose a motion representation for single-person motion generation that incorporates ground contact information and motion features. They define a pose $\mathbf{p}$ as a 9-tuple of $(r^a, r^x, r^z, r^y, \boldsymbol{j}^p, \boldsymbol{j}^v, \boldsymbol{j}^r, \boldsymbol{c}^f)$ where, $r^a \in \mathbb{R}$ is root angular velocity along the Y-axis; $(r^x, r^z \in \mathbb{R})$ are root linear velocities along the XZ-plane; $r^y \in \mathbb{R}$ is root height; $\boldsymbol{j}^p \in \mathbb{R}^{3j}$, $\boldsymbol{j}^v \in \mathbb{R}^{3j}$, $\boldsymbol{j}^r \in \mathbb{R}^{6j}$ are the local joint positions, velocities and rotations in the root space where j is the number of joints; $\boldsymbol{c}^f \in \mathbb{R}^4$ are binary foot contact features for the heel and toe joints. This representation has been used with success [20, 70], but it cannot be applied directly to multi-person scenarios as relative global spatial information is lost.

Liang et al. [19] instead introduce global relative rotation and translation features into the pose $\mathbf{p}$. They find that the global absolute trajectories derived from this representation tend to drift. While not as important for short-sequence single-person generation where local motion is prioritized, the potential accumulation for multi-person synthesis could result in highly implausible spatial relations.

Liang et al. propose a non-canonical motion representation where they directly include the global trajectories of both individuals within the same world frame. The idea is that by keeping the rotations and velocties within the world frame, they can easily extract the global relative rotation and translation through IK and the root position which helps to avoid drift. For the purpose of interactive partner dancing, we utilize this same non-canonical motion representation for DuoDance where the i-th motion state for person a $x_a{}^i$ is defined as follows:

$$x_a{}^i = (j_{a_g}{}^p, j_{a_g}{}^v, j_a{}^r, c_a{}^f), \tag{3.1}$$

where global joint positions $j_{a_g}{}^p \in \mathbb{R}^{3N_j}$ and global joint velocities $j_{a_g}{}^v \in \mathbb{R}^{3N_j}$ are in the world frame, local 6D representations of joint rotations $j_a{}^r \in \mathbb{R}^{6N_j}$ are in the root frame, $c_a{}^f \in \mathbb{R}^4$ are binary foot-ground contact features. When examining the results of DuoDance, we only use the global joint positions for quantitative and qualitative assessment.

3.2 Audio Representation

Our goal is to synthesize motion from audio, therefore the decision of which audio representation to use as conditioning input for DuoDance may be a determining factor in achieving high-quality results. In the text-to-image domain, results have shown that scaling the size of the text-encoder improves sample quality significantly more than scaling the size of the diffusion model [93]. The conditional representation and encoding for conditional diffusion models need to be carefully chosen. Learning representative features of audio can be more computationally challenging than that of the text or image domain when considering the high sampling rates of audio. While not a problem for short clips, as input length increases so does the need to accurately capture the wider array of musical structures. EDGE [5] showed that users favored results generated with Jukebox features over the "baseline" features of FACT [3]. MoFusion [8] utilizes Mel spectrograms for their music-to-motion generation with the claim their model could theoretically use speech instead of music to generate motion. As we focus solely on music as our conditioning for DuoDance, we will compare the "baseline", Mel spectrogram, and Jukebox representations to see which generates user-preferred motion. Both "baseline" and Mel spectrogram will be extracted from the Librosa toolbox [94] as it is a

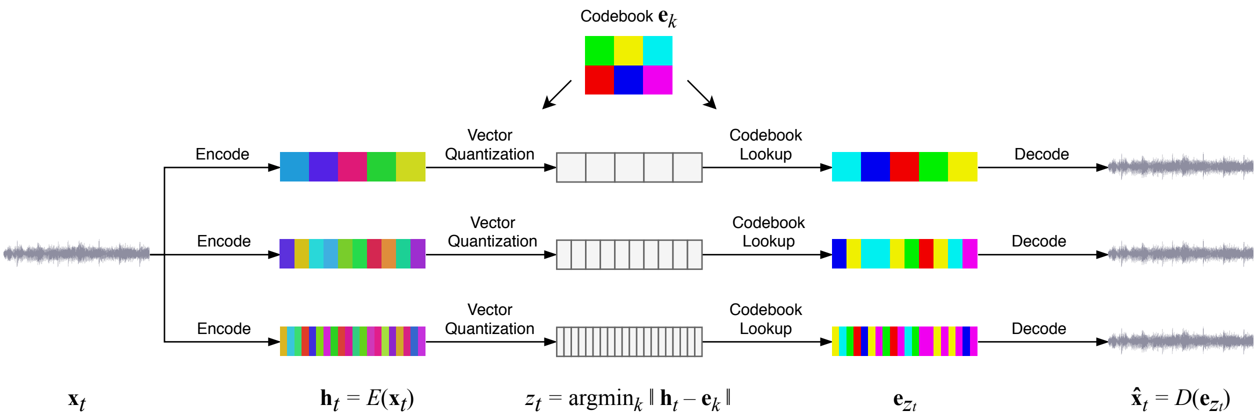


Figure 8: Training for the three VQ-VAE models with different temporal resolutions. [19].

common framework used by several audio-driven dance generation methods [3]. To extract Jukebox features, Tseng et al. [5] provide a memory-efficient implementation to extract audio representations.

3.2.1 Baseline Features

The baseline representation B_t at each frame t is defined as follows:

$$B_t = (e_t, M_t, C_t, p_t, b_t), \tag{3.2}$$

where $e_t \in \mathbb{R}$ is onset strength at frame t; $M_t \in \mathbb{R}^{20}$ are Mel-Frequency Cepstral Coefficients (MFCCs); $C_t \in \mathbb{R}^12$ are Chroma Energy Normalised Statistics (CENS); $p_t \in 0, 1$ is a binary indicator of a peak within the envelope at time t; $b_t \in 0, 1$ is a binary indicator of a beat at time t.
For onset strength calculations, a log-power Mel spectrogram S is generated as well as a reference spectrogram R. R is generated by applying a local maximum filter along the frequency axis of S. The spectral flux is calculated as the difference between S and R for each frequency bin. This difference shows increases in energy, which can represent the start of a note or onset. MFCCs provide a simplified representation of the timbre of the audio. Chromograms are used to analyze the pitch of the audio. Chroma CENS represent statistics taken over a short-time period within chroma energy bands. They are robust to dynamics, timbre, and articulation and are often used in audio matching and retrieval applications [94]. The binary peak and beat indicators inform the model when significant events occur in the music. These are especially useful for trying to match motion with the rhythm of the audio. Assuming a rate of 30 FPS, the resulting feature vector using baseline features is (s x 30 x 35) where s is the audio length in seconds.

3.2.2 Mel Spectrogram Features

A mel-frequency spectrogram is obtained by applying a nonlinear transform to the frequency axis of the short-time Fourier transform (STFT). This summarizes the frequency content with fewer dimensions. By emphasizing details in lower frequencies and de-emphasizing the details in higher frequencies, they help to model speech intelligibility and have seen usage in speech recognition for decades [95]. Their applications in music-to-motion are limited and therefore a comparative analysis should be performed. Assuming a rate of 30FPS, the resulting feature vector using a mel-frequency spectrogram is (s x 30 x 80) where s is the audio length in seconds.

3.2.3 Jukebox Features

Jukebox [76] is a music generative model pre-trained on codified audio from 1.2M songs of varying artists and genres. It utilizes a hierarchical VQ-VAE model to compress the raw audio to a lower-dimensional space at three different levels of abstraction Figure 8. An Encoder $E(\mathbf{x}_t)$ encodes the input $\mathbf{x}_t$ into a sequence of latent vectors $\mathbf{h}_t$ which are then quantized into discrete tokens z_t. These are then mapped to the nearest vector

$\mathbf{e}_{Z_t}$ from a codebook. Finally a decoder $D(\mathbf{e}_{Z_t})$ decodes the embedding vectors back to the input space. Random restarts are used whenever the mean usage of a codebook vector dips below a specified threshold, allowing the VQ-VAE to avoid codebook collapse. Next, a prior over the compressed space is learned using a cascade of Transformers with sparse attention. Each Transformer model takes conditioning information such as genre, artist, timing, and lyrics. When generating music they find their samples are musically coherent, diverse, and representative of the conditioned genre and artist.

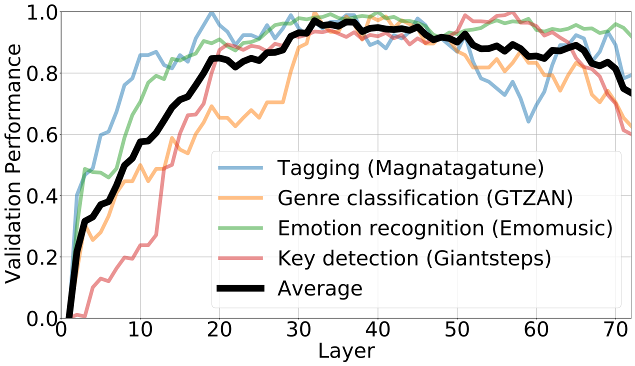


Figure 9: Performance of downstream MIR tasks for each Jukebox layer [83].

Castellon et al. [83] investigate whether Jukebox's codified audio language modeling (CALM) learns representations that are useful for downstream MIR tasks. The 5B parameter model without lyrics conditioning is a sparse transformer containing 72 layers. Each layer contains 4800-dimensional activations for each sample in the codified audio sequence, approximately 345 per second. This results in almost 2.4GB of information for a single 5-second clip when represented as a 32-bit floating point, 5 x 345 x 72 x 4800 x 32. They reduce the amount of data by mean pooling across time to achieve (72x4800) features for any length audio sequence. To further reduce the computation cost, they select only one of the 72 layers to obtain 4800 features. By training linear models using representations for different layers on varying MIR tasks, they conclude that the middle layers contain the strongest representations (see Figure 9). They compare the representations of jukebox to hand-crafted features such as MFCCs and Chroma. Results indicate probing features from a CALM pre-trained model like jukebox results in better performance across all MIR tasks.
For audio-to-motion generation where each frame of audio is paired with a motion frame, mean pooling would result in the loss of temporal information. Similar to Tseng et al. [5], we extract our jukebox representations and downsample the extracted jukebox features to 30 FPS to match the motion data frame rate. We deviate slightly by selecting the 36th Layer instead of the 66th. Considering the findings of Castellon et al. [83], it is unclear why the authors chose a different set of activations for EDGE. The resulting feature vector using jukebox features is (s x 30 x 4800) where s is the audio length in seconds.

3.3 DuoDance Diffusion Framework

While there exist numerous frameworks for the task of generative synthesis, diffusion models have gained significant popularity recently as they've been shown to outperform other methods for tasks such as image synthesis [69] and music-to-dance synthesis [8]. We adapt the InterGen [19] framework to generate motion for two dancers. InterGen is the current SOTA model for the generation of interactive two-person motion. The following definitions and explanations are based on [19, 96]. For a general overview of the DuoDance architecture, please refer to

3.3.1 Diffusion Process

Let $p_0(x)$ be the motion distribution of human interaction data, and let p_T be a standard Gaussian distribution. During the forward process, time-dependent i.i.d Gaussian noise is artificially inserted into the samples of p_0. This gives a diffusion process $p_t(x)_{t=0}^{T}$ indexed by a continuous time variable $t \in [0, T]$. We can obtain samples $x^{(0)} \sim p_0$ by reversing the process starting with samples of $x^{(T)} \sim p_T$. The reverse of a diffusion

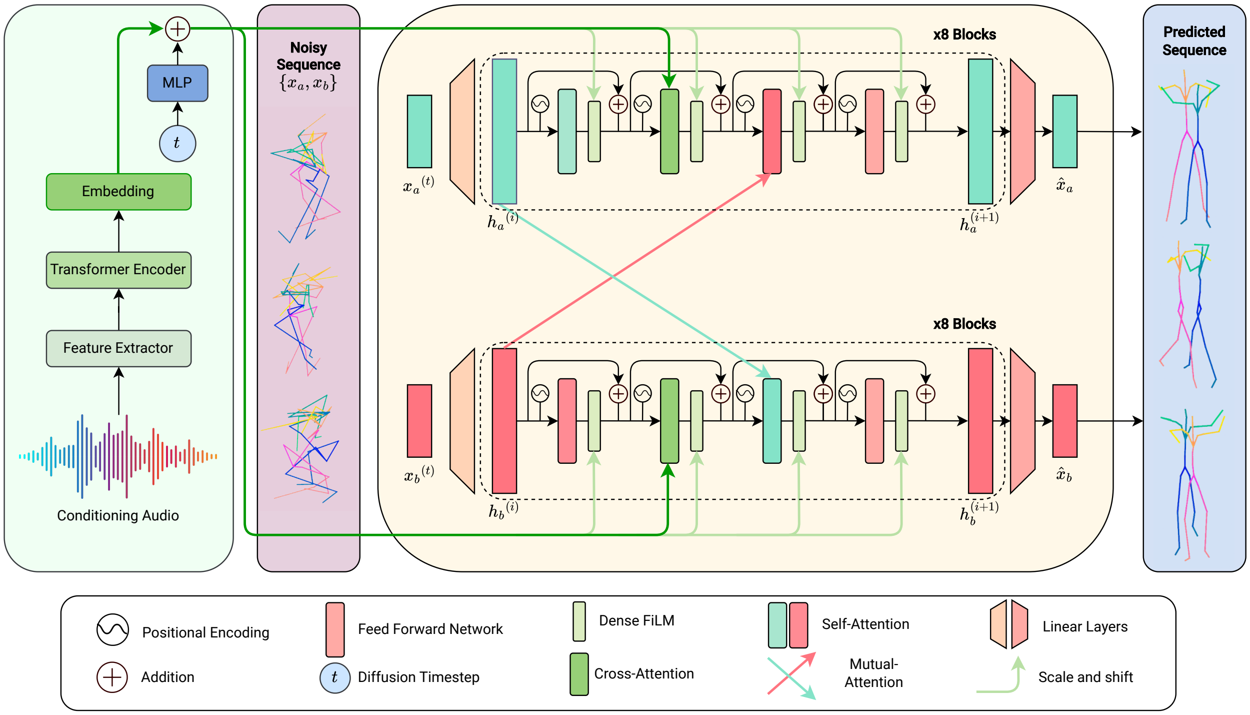


Figure 10: The overview of proposed DuoDance model. We introduce a cross-attention mechanism between self-attention and mutual-attention layers to strengthen our conditioning signal. In addition, we incorporate rotary positional embeddings to better capture the sequential nature of dance movements.

process is also a diffusion process running backward in time given by the reverse time Stochastic Differential Equation (SDE):

$$d\mathbf{x} = \left[\mathbf{f}(\mathbf{x}, t) - \sigma_t^2 \nabla_{\mathbf{x}} \log p_t(\mathbf{x})\right] dt + \sigma_t d\mathbf{w}, \tag{3.3}$$

,

where $\mathbf{f}(\cdot, t) : \mathbb{R}^d \to \mathbb{R}^d$ is the drift coefficient of $x(t)$, $\sigma_t^2 : \mathbb{R}^d \to \mathbb{R}^d$ is the diffusion coefficient that increases over time to control the noise level, dt is an infinitesimal negative timestep, $\mathbf{w}$ is a standard Weiner process when time flows backward from T to 0, and $\nabla_{\mathbf{x}} \log p_t(\mathbf{x})$ is the score function of each marginal distribution. Once the score function for all t is known, we can derive the reverse diffusion process from Equation 3.3 and simulate it to sample from p_0. This can be obtained from the expectation of x given $x^{(t)}$:

$$\nabla_{\mathbf{x}^{(t)}} \log p_t \left(\mathbf{x}^{(t)}\right) = \left(\mathbb{E}\left[\mathbf{x} \mid \mathbf{x}^{(t)}\right] - \mathbf{x}^{(t)}\right) / \sigma_t^2. \tag{3.4}$$

There exists a deterministic process that shares the same marginal distribution $p_t(x)_{t=0}^T$ as the SDE. This process satisfies an ordinary differential equation (ODE) which is obtained by dropping the noise term in Equation 3.3. Named the probability flow ODE, this ODE can accelerate the sampling process by performing a linear interpolation between $x^{(t)}$ and $(\mathbb{E}\left[\mathbf{x} \mid \mathbf{x}^{(t)}\right]$.

3.3.2 Interaction Diffusion

Liang et al. [19] assume that the interaction between two people is symmetric, i.e., the order of motion does not change the semantics of the interaction itself. The interaction between person a and person b $\{x_a, x_b\}$ is equivalent to $\{x_b, x_a\}$. This holds true for partner dancing, despite the notion of a leader and follower. Person a can lead while person b follows, and person b can lead while person a follows. Therefore, two people share the same single-person motion marginal distribution:

$$p\left(\mathbf{x}_a, \mathbf{x}_b\right) \equiv p\left(\mathbf{x}_b, \mathbf{x}_a\right). \tag{3.5}$$

Given noise is independent of the data distribution, at any time t the observations of a and b's noisy motion can be treated equivalently:

$$p_t\left(\mathbf{x}_a^{(t)}, \mathbf{x}_b^{(t)}\right) \equiv p_t\left(\mathbf{x}_b^{(t)} \mathbf{x}_a^{(t)}\right) \tag{3.6}$$

Based on the above equation the score function can be formulated as follows:

$$\begin{aligned} & \nabla_{\mathbf{x}^{(t)}} \log p_t\left(\mathbf{x}^{(t)}\right) \\ & =\left[\nabla_{\mathbf{x}_a^{(t)}} \log p_t\left(\mathbf{x}_a^{(t)}, \mathbf{x}_b^{(t)}\right), \nabla_{\mathbf{x}_b^{(t)}} \log p_t\left(\mathbf{x}_a^{(t)}, \mathbf{x}_b^{(t)}\right)\right] \\ & =\left[\nabla_{\mathbf{x}_a^{(t)}} \log p_t\left(\mathbf{x}_a^{(t)}, \mathbf{x}_b^{(t)}\right), \nabla_{\mathbf{x}_b^{(t)}} \log p_t\left(\mathbf{x}_b^{(t)}, \mathbf{x}_a^{(t)}\right)\right] \end{aligned} \tag{3.7}$$

This allows us to use the same network to denoise the motion of person a and person b. The denoising process aims to predict the original motion from the noisy motion and requires some loss function to minimize the difference between the two. We apply the same loss function $\mathcal{L}_{\text{simple}}$ as InterGen and is defined below:

$$\begin{aligned} \mathcal{L}_{\text{simple }} = \mathbb{E}_{\mathbf{x}, t, \epsilon} & \left[\lambda_t\left\|\mathbf{x}_a-D_\theta\left(\mathbf{x}_a+\sigma_t \epsilon_a, \mathbf{x}_b+\sigma_t \epsilon_b, t, c\right)\right\|_2^2\right. \\ & \left.+\lambda_t\left\|\mathbf{x}_b-D_\theta\left(\mathbf{x}_b+\sigma_t \epsilon_b, \mathbf{x}_a+\sigma_t \epsilon_a, t, c\right)\right\|_2^2\right] \end{aligned} \tag{3.8}$$

Here D_θ refers to the denoisers that share the common network weights. The input consists of its own noisy motion to denoise, the cooperator's noisy motion containing the other person's motion, the time t, and the condition c (text for InterGen, and audio for DuoDance). The noise $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and λ_t is the loss weighting factor (see Section 3.3.5).

3.3.3 Diffusion Architecture

We propose a novel architecture built on the existing network structure of InterGen [19] and reference the same formulations. The noisy motion is first embedded into a common latent space using a linear transformation. This embedding is positionally encoded into the hidden states $h^{(0)}$ where it is processed by N attention-based blocks to obtain the denoised hidden state $h^{(N)}$. These blocks are made up of a self-attention layer, a cross-attention layer, a mutual-attention layer, and finally, a feed-forward network. The self-attention layer processes the block's own hidden states $h^{(i)}$ to generate a context vector $c^{(i)}$. For person a, the context vector $c_a^{(i)}$ is formulated as follows:

$$\begin{gathered} \mathbf{c}_a^{(i)}=\operatorname{Attn}(\mathbf{Q}, \mathbf{K}, \mathbf{V})=\operatorname{softmax}\left(\frac{\mathbf{Q K}^T}{\sqrt{C}}\right) \mathbf{V} \\ \mathbf{Q}=\mathbf{h}_a^{(i)} \mathbf{W}_s^Q, \mathbf{K}=\mathbf{h}_a^{(i)} \mathbf{W}_s^K, \mathbf{V}=\mathbf{h}_a^{(i)} \mathbf{W}_s^V, \end{gathered} \tag{3.9}$$

where C is the number of channels in the attention layer and W_s are trainable weights. For person b, $c_b^{(i)}$ is calculated in the same way using shared weights W_s. We introduce a cross-attention layer between the self-attention and mutual-attention layers to enforce conditional generation. This cross-attention layer processes the context vector $c^{(i)}$ and the hidden state of the conditioning input $h_c(i)$ to generate a new context vector $c^{(i)\prime}$:

$$\begin{aligned} & \mathbf{c}_a^{(i')}=\operatorname{Attn}(\mathbf{Q_c}, \mathbf{K_c}, \mathbf{V_c}) \\ & \mathbf{Q_c}=\mathbf{c}_a^{(i)} \mathbf{W}_c^Q, \mathbf{K}=\mathbf{h}_c^{(i)} \mathbf{W}_c^K, \mathbf{V}=\mathbf{h}_c^{(i)} \mathbf{W}_c^V, \end{aligned} \tag{3.10}$$

where W_c are trainable weights specific to the motion-audio cross-attention mechanism. We find this addition to be key in strengthening the signal of the conditioning (see autoref results section).

Unlike the previous self-attention and cross-attention mechanisms, mutual-attention allows for inter-block information exchange. Each block utilizes the context vectors derived from its counterpart to update its own hidden state. This cross-block interaction assists in the model's learning of interdependencies between the data. The computation of the next hidden state $h^{(i+1)}$ is calculated as follows:

$$\begin{aligned}
&\mathbf{h}_a^{(i+1)} = FF\left(\operatorname{Attn}\left(\mathbf{Q}_a, \mathbf{K}_b, \mathbf{V}_b\right)\right), \\
&\mathbf{h}_b^{(i+1)} = FF\left(\operatorname{Attn}\left(\mathbf{Q}_b, \mathbf{K}_a, \mathbf{V}_a\right)\right), \\
&\mathbf{Q}_a = \mathbf{c}_a^{(i)} \mathbf{W}_m^Q, \mathbf{K}_a = \mathbf{h}_a^{(i)} \mathbf{W}_m^K, \mathbf{V}_a = \mathbf{h}_a^{(i)} \mathbf{W}_m^V, \\
&\mathbf{Q}_b = \mathbf{c}_b^{(i)} \mathbf{W}_m^Q, \mathbf{K}_b = \mathbf{h}_b^{(i)} \mathbf{W}_m^K, \mathbf{V}_b = \mathbf{h}_b^{(i)} \mathbf{W}_m^V,
\end{aligned} \tag{3.11}$$

where $\mathbf{W}_m$ are shared trainable weights. Adaptive layer normalization is replaced in favor of Feature-wise Linear Modulation (FiLM) [97] after each layer. Timestep information is positionally encoded via the FiLM layers and used to scale and shift the output of each layer.

3.3.4 Conditional Encoding

We adapt the InterGen model to condition motion on audio instead of text. Originally, a frozen CLIP-ViTL/14 [98] model is used to create the text embeddings which are then linearly transformed into the common latent space. We replace the CLIP model and employ a transformer encoder architecture similar to EDGE [5] to encode the music. This consists of multi-head self-attention and feed-forward layers to capture the temporal dependencies of the rhythm, pitch, and timbre of the audio. The diffusion timesteps are encoded and passed through a FFN network to be concatenated to the encoded audio.

3.3.5 Losses

We apply the same regularization losses as InterGen given that dancing is a subset of general human interaction. Adopting losses specific to the case of partner dancing remains a challenging task given the diverse nature of the movement across different styles. We define the foot contact loss $\mathcal{L}_{\text{foot}}$ originally from Tevet et al. [70] without forward kinematics FK as:

$$\mathcal{L}_{\text{foot}} = \frac{1}{N-1} \sum_{i=1}^{N-1} \left\| \left(\hat{x}_{\text{foot}}^{i+1} - \hat{x}_{\text{foot}}^{i} \right) \cdot f_i \right\|_2^2, \tag{3.12}$$

where $\hat{x}_{\text{foot}}^{i}$ are the predicted foot positions at frame i and f_i is a binary foot contact mask $f_i \in 0, 1$. This loss is designed to prevent foot sliding by enforcing minimal foot movement when there is ground contact. Tevet et al. [70] find omitting this loss contributes to artifacts such as shakiness and unnatural movement. The joint velocity loss $\mathcal{L}_{\text{vel}}$ helps to enforce physical plausibility penalizing rapid or unnatural changes in joint positions across frames:

$$\mathcal{L}_{\text{vel}} = \frac{1}{N-1} \sum_{i=1}^{N-1} \left\| \left(x_0^{i+1} - x_0^{i} \right) - \left(\hat{x}_0^{i+1} - \hat{x}_0^{i} \right) \right\|_2^2. \tag{3.13}$$

As we predict joint positions from the non-canonical motion representation, there needs to be a constraint on the global joint positions to ensure skeletal consistency. InterGen proposes the bone length loss $\mathcal{L}_{BL}$:

$$\mathcal{L}_{BL} = \left\| B\left(\hat{\mathbf{x}}_a\right) - B\left(\mathbf{x}_a\right) \right\|_2^2 + \left\| B\left(\hat{\mathbf{x}}_b\right) - B\left(\mathbf{x}_b\right) \right\|_2^2, \tag{3.14}$$

where the global joint positions of $\mathbf{x}$ are used to calculate bone lengths B from a pre-defined human body kinematic tree.

Liang et al.[19] introduce two novel interactive losses. The first is masked joint distance map (DM) loss which measures the $N_j \times N_j$ joint distance map of two people and compares this to the ground truth only when they are physically close:

$$\mathcal{L}_{DM} = \left\| \left(M\left(\hat{\mathbf{x}}_a, \hat{\mathbf{x}}_b\right) - M\left(\mathbf{x}_a, \mathbf{x}_b\right) \right) \odot I\left(M_{xz}\left(\mathbf{x}_a, \mathbf{x}_b\right) < \bar{M} \right) \right\|_2^2. \tag{3.15}$$

Whenever the distance between the two people on the XZ-plane (M_{xz}) is smaller than some threshold ($\bar{M}$), the Indicator function $I(\cdot)$ activates the loss between the joint distance maps (M) of the two people through the Hadamard product $\odot$. As physical interaction between two people requires them to be in close proximity, this loss allows the model to ignore irrelevant features when far apart. The second novel interactive loss is the Relative Orientation (RO) loss:

$$\mathcal{L}_{RO} = \left\| O\left(IK\left(\hat{\mathbf{x}}_a\right), \mathrm{IK}\left(\hat{\mathbf{x}}_b\right)\right) - O\left(IK\left(\mathbf{x}_a\right), IK\left(\mathbf{x}_b\right)\right)\right\|_2^2 \tag{3.16}$$

This loss approximates the relative orientation of two people and aligns it with the ground truth. O is the 2D relative orientation around the Y-axis, and $IK(\cdot)$ is the inverse kinematics process to obtain the person's joint rotations.

The final regularization loss can be defined as follows:

$$\mathcal{L}_{\text{reg}} = \lambda_{\text{vel}} \mathcal{L}_{\text{vel}} + \lambda_{\text{foot}} \mathcal{L}_{\text{foot}} + \lambda_{BL}\mathcal{L}_{BL} + \lambda_{DM}\mathcal{L}_{DM} + \lambda_{RO}\mathcal{L}_{RO}, \tag{3.17}$$

where $\lambda vel = 30$, $\lambda_{foot} = 30$, $\lambda_{BL} = 10$, $\lambda_{DM} = 3$, and $\lambda_{RO} = 0.01$. These λ hyperparameters are the same weights used by Liang et al. and are carefully calibrated to regulate the magnitude orders of their corresponding terms [19]. This ensures a balanced contribution from each loss term.

4 Experimental Setup

We lay out the necessary steps to evaluate the performance of DuoDance. First, we define the construction and augmentations of the dataset used for training and testing. Next, we select and reason our choice of evaluation metrics for both quantitative and qualitative assessment. Finally, we provide precise implementation details relating to the training and inference of DuoDance.

4.1 Datasets

As there is no existing public audio-partner dance paired dataset, we adapt the InterHuman dataset from Liang et. al [19]. The InterHuman dataset contains a subset of dancing sequences featuring multiple dance genres. We manually select all motion descriptions which include the following words: "danc(e/ing)", "latin", "turn(ing)", "music", and "sway(ing)". We then visualize the motion sequence to ensure its suitability for the task. Duplicates or subdivisions are removed in favor of the original motion. Clips that include performers practicing and not moving rhythmically are also removed. This culling results in a total of 35.6 minutes of whole-body motion of two people dancing. The evaluation of suitable motions and the classification of the dance genre are assisted by a dance expert with over 20 years of expertise.

Diffusion models benefit from large amounts of data so we artificially inflate the dataset by pairing multiple pieces of audio to the same motion sequence. Each motion sequence is manually aligned using video-editing software with three different sections from the same audio clip. This process is repeated multiple times for the same motion sequence and different audio clips. Once aligned, the aligned audio clips are exported to wav files. Each audio clip is aligned with at least two different motions as there are a myriad of ways dancers may interpret the music on which they dance. This should ensure the model does not learn to associate specific audio with a singular motion. The leader and follower's order in the motion data is not always consistent. Therefore during training for each model, we randomly decide to swap the order of the motions. This is in line with the assumption that the interaction between two people is symmetric (Equation 3.5). Swapping the motion ensures the model does not learn to associate specific movements only with a single person. Either the first or second person can be the leader or follower therefore the ordering isn't important.

4.1.1 Dataset Split

The final size of our expanded dataset is 316.62 minutes. Each motion and paired audio clip is cut to 5 seconds with 30FPS. Any clips shorter than 5 seconds are null padded. We choose 5-second clips as it is enough time to capture essential movements and audio features that are representative of the varying genres. Choosing 5-second clips over 10 also reduces the chance of overfitting by introducing more varied clips. Our dataset consists of nine different dancing genres as shown in Figure 11). To ensure the same percentage of samples from dance genres across the train and test splits, we manually apply a stratified sampling technique. All 5-second clips from the original sequence are grouped into one split to prevent data leakage. Similarly, we ensure no single audio is used in both the train and test splits. We utilize an 80% to 20% split for training and testing.

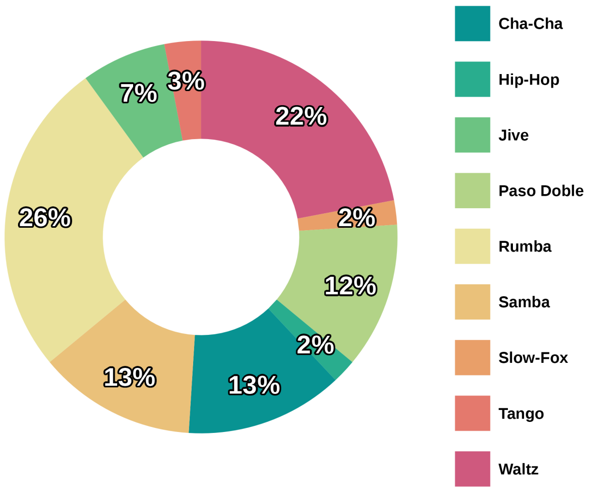


Figure 11: Distribution in % of dance genres in the finalised dataset.

4.1.2 Data Augmentation

We choose to measure the effect of data augmentation on our model's performance. First, we ensure each clip has a 30-frame overlap with the next. This means that the first 30 frames of each subsequent clip are identical to the last 30 frames of the clip prior. We also double the dataset by mirroring the motions. Typically, partner dancing often has specific sides that are important in maintaining the style of the genre. For example, when dancing Salsa "On-1", the leader steps forward with their left foot while the follower steps back with their right. When mirroring the motion, the leader would step forward with their right, and the follower back with their left. While these nuances are important to consider for creating "accurate" motion, we theorize mirrored motions will improve the model's capacity to learn general motion patterns and will not be negatively assessed during a user preference study.

4.1.3 Dataset List

We decide to train our model on subsets of the data to see the effects of data availability and data consistency. For the rest of section 4 we will assign the following names to these variants:

- **DuoDance-20** - DuoDance model trained on 20% of the unaugmented data using Jukebox features.
- **DuoDance-50** - DuoDance model trained on 50% of the unaugmented data using Jukebox features.
- **DuoDance-100** - DuoDance model trained on 100% of the unaugmented data using Jukebox features.
- **DuoDance-Baseline** - DuoDance model trained on 100% of the unaugmented data using the librosa features.
- **DuoDance-Mel** - DuoDance model trained on 100% of the unaugmented data using the mel spectrogram features.
- **DuoDance-Waltz** - DuoDance model trained on only unaugmented Waltz motion using jukebox features (accounting for roughly 22% of the overall data).

- **DuoDance-Aug** - DuoDance model trained on 100% of the augmented data using Jukebox features.

By limiting the amount of data available to our model, we can evaluate the model's performance as more data is introduced.

4.2 Evalutation Metrics

We deviate from InterGen and look towards other audio-to-motion methods when choosing metrics to evaluate our results. In text-to-motion generation, a motion feature extractor and text feature extractor are trained under contrastive loss to ensure geometrically close feature vectors [19, 32]. We attempt to follow suit by training a motion feature extractor and audio feature extractor under contrastive loss. However, this was unsuccessful likely in part due to pairing multiple audios to the same motion and vice-versa. Instead, we decide to apply the following commonly used metrics for our task:

Frechet Inception Distances (FID): Attempts have been made to establish a reliable motion encoder for computing FID across real and generated motion data [99]. However, it has been shown that these attempts fail to capture important features like foot sliding degradation [100]. We decide to examine the geometric (g) and kinetic (k) features of the motion similar to [3, 82] to compute FID_g and FID_k. Typically, a lower FID is indicative of better performance. We investigate whether the findings of Li et al. [5] regarding the suitability of these metrics are applicable to our generated motion.

Motion Diversity: Motion diversity is the average feature distance across the set of generated motions. We randomly sample 100 pairs of motion and compute Div_g and Div_k for the geometric and kinetic feature distributions respectively. These values should match those of the real data as the goal is to create motion resembling the ground truth distribution.

Multimodality: As diffusion models are nondeterministic, our model should be capable of generating a diverse range of dances for the same audio input. We compute the multimodality metrics of DuoDance (MM_g and MM_k) by computing average FID_g and FID_k between sampled motions generated from the same audio. The average over all audio samples is reported.

Physical Foot Contact: Following EDGE, we compute the Physical Foot Contact (PFC) score which is intended to measure physical plausibility. We compute the PFC for person 1 and person 2 individually and report the average across all test data. We examine the robustness of this metric in Section 5.0.1.

User Study: We utilise an A/B testing strategy to determine user preference in terms of physical plausibility, motion complexity, and music-motion correlation. For this study we use rendered videos of the generated motion from different model variations for the same audio clip. The participants are presented with two randomly ordered videos and asked to watch and evaluate the motion. We ask the participants the following questions:

1. Which dance is more appropriate for the given music?
2. Which dance is more physically plausible?
3. Which dance is more complex and varied?
4. Which dance do you prefer?

While previous works have utilised user preference surveys [3, 5], they fail to capture which elements may influence the user's decision. These questions allow us to subjectively assess our results in terms of physical plausibility, complexity, and music-motion correlation. We also include comparisons to the ground truth data. This survey is conducted using Google Forms where users are asked to compare 28 pairs of motion, each 10 seconds long. We follow EDGE [5] in choosing a longer sample duration as it provides more information for the user to interpret the motion, particularly necessary for those with limited knowledge of the domain. From these results, we can compute the relative quality of our models by computing each model's

Hyperparameters	DuoDance
Optimizer	Adam
Learning Rate	$1e^{-4}$
Number of Epochs	1500
Diffusion Steps	1000
β schedule	cosine
Motion Duration	5 seconds
Motion FPS	30
Motion Dimensions	262
Audio Dimensions	35(B); 80(M); 4800(J)
Classifier Free Dropout	0.25
Num Heads	8
Num Layers	8
Transformer Dimension	1024
Feed Forward Dimension	2048
Dropout	0.1

Table 1: Hyperparameter values of our proposed DuoDance model. The size of the audio dimensions varies depending on the type of representation where (B), (M), and (J) correspond to “Baseline”, Mel Spectrogram, and Jukebox features respectively.

Elo ranking similarly to Tseng et al. [5].

Beat Align Score: We choose to not measure the Beat Align Score (BAS) for our evaluation. Typically, this metric is used to determine how in-sync the movement is with the music. This is done by rewarding kinematic beats (local joint minima) that align with music beats. Li. et al [3] measure the distance per kinematic beat, while Siyao et al. [82] measure the distance per audio beat. In their evaluation of EDGE, Tseng et al. [5] find their model outperforms ground truth motion in terms of BAS. They reason that while this metric has served to drive progress in the field, it is no longer suitable for complex dance generation. This is because dancers interpret the structure of the music in different ways and may choose to adjust their motion to the move on the lyrics, different instruments, or experiment with timing. As such, a rigid metric that penalises creativity does not reflect the quality of the motion.

4.3 Implementation Details

We train DuoDance for 1,500 epochs with a batch size of 64. We use the same training configuration as InterGen, where each model contains $N = 8$ blocks whose latent dimension is 1024, and each attention layer has 8 heads. All models use an AdamW [101] optimizer with betas of (0.9, 0.999), a weight decay of 2×10^{-5}, a maximum learning rate of 10^{-4}, and a cosine LR schedule with 10 linear warm-up epochs [19]. During training, the number of diffusion timesteps is set to 1,000, and during sampling we apply the DDIM [67] strategy with 50 timesteps. For the audio encoder, we use a transformer encoder with $N = 2$ layers, a dropout rate of 0.1, and 8 parallel attention heads. We train DuoDance using classifier-free guidance by randomly setting 25% of the audio embeddings to $\emptyset$. During sampling, the guidance coefficient is set to 2. Please refer to Table 1 for additional information.

4.3.1 Long-Form Sampling

In order to generate coherent samples of 10 seconds for the user preference survey, we take inspiration from EDGE [5] to generate sequences of arbitrary length. This is achieved by generating a batch of 5-second sequences simultaneously during inference. After each timestep, the first 2.5 seconds of a sequence are temporally constrained to match the last 2.5 seconds of the previous sequence. The conditioning input has a 2.5-second overlap for each sequence in the batch. We interpolate between two overlapping motion slices with a linearly decaying weight. This approach is suitable for generating consistent and smooth motion.

5 Discussion and Results

For all quantitative metrics, we report the average result for the test data across three iterations. These values can be seen in Table 2. First, we discuss the limitations of the Physical Foot Contact score. Next, we compare the results of DuoDance-20 and DuoDance-Waltz. We then analyse the impact of data availability and augmentation. We discuss the impact of audio representation on performance, before moving to the user survey. Finally, we briefly mention the impact of our cross-attention mechanism in relation to InterGen.

5.0.1 Physical Foot Contact Score

The Physical Foot Contact (PFC) metric introduced by Tseng et al. [5], was designed to evaluate physical plausibility. The metric arises from the following observations:

1. Any center of mass (COM) acceleration on the horizontal planer must be due to static contact between the foot and the ground.
2. On the vertical axis, any positive COM acceleration must be due to static foot contact.

The adherence of these observations can be formulated as the average over time of the following:

$$s^i = \left\|\overline{\boldsymbol{a}}^i_{\text{COM}}\right\| \cdot \left\|\mathbf{v}^i_{\text{Left Foot}}\right\| \cdot \left\|\mathbf{v}^i_{\text{Right Foot}}\right\| \tag{9}$$

$$PFC = \frac{1}{N \cdot \max_{1\leq j\leq N}\left\|\overline{\boldsymbol{a}}^j_{\text{COM}}\right\|}\sum_{i=1}^{N} s^i \tag{10}$$

where $\overline{\boldsymbol{a}}^i_{\text{COM}}$ consists of the acceleration across the x, y, and z dimensions. Despite their findings, we see that the PFC score is not a suitable evaluation metric for our generated motion. Looking at Table 2, all models have significantly lower PFC scores than the ground truth. This indicates the metric cannot capture physical plausibility within the specific motion distribution. This score has one fundamental flaw for use in motion generation. It assumes a constant velocity (or zero acceleration) is indicative of realistic motion. Any semblance of motion drifting where the motion slides across the horizontal plane at a constant rate would result in a perfect score under this metric. As our dataset consists of motion with varying movement across this plane, it is likely that small amounts of motion drift contribute to the low PFC scores. Furthermore, both observations do not hold when applied to partner dancing. As there can be additional forces enacted upon one dancer from another, this metric would consider motion such as a lift to be physically implausible. While these types of motions are not present in our dataset, it is worth noting for future considerations. For the remainder of this section, we do not compare the PFC results between models.

5.1 DuoDance-20 vs. DuoDance-Waltz

Our user study shows that users prefer the motion of DuoDance-Waltz over DuoDance-20. It is important to note that the comparison was made using motion generated from waltz audio. The motion of DuoDance-Waltz was found to be more physically plausible and better suited for the given audio. DuoDance-Waltz produces a higher FID$_k$ for kinetic features while FID$_d$ is lower. Visually inspecting the generated motion, we see much more horizontal movement vs. DuoDance-20. As the waltz is a dynamic dance where large movement across the horizontal plane is common, it is possible DuoDance-Waltz focuses on these grand movements while putting less importance on the smaller, subtle movements when standing in place. When considering the geometric feature distance, we find our model trained solely on waltz motion is closer to the ground truth than our model trained on a variety of genres. The “frame” of dancers used during the waltz, i.e. the positioning of the upper torso and arms, is normally consistent throughout the entire dance. Typically, the leader has their right arm wrapped around the left shoulder blade of the follower. The follower rests their left arm on the leader’s right arm. The leader and follower connect their left and right hands respectively, and hold them slightly elevated. When examining the output of DuoDance-Waltz, this frame is consistent throughout many of the generated motions. See Figure 12 for a visual representation of the dancer’s frame. Despite the added variance of multiple genres in DuoDance-20, we achieve lower MM$_k$ and MM$_d$ scores. This

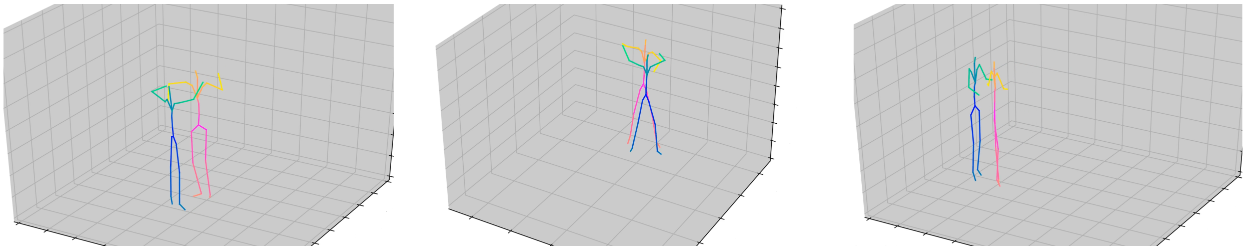


Figure 12: From left to right: DuoDance-Waltz, Ground Truth, DuoDance-20. We can see that DuoDance-Waltz more closely resembles the dancer's frame of the ground truth motion. DuoDance-20 produces motion where arm positioning is more varied. This may contribute to a lower FID$_g$ score for DuoDance-Waltz.

could be a result of failing to learn the conditioning influence and instead generating motion that fits general motion distribution.

Datasets	Elo ↑	FID$_k$ ↓	FID$_g$ ↓	Div$_k$ →	Div$_g$ →	MM$_k$ ↑	MM$_g$ ↑	PFC ↓
Grount Truth (20%)	-	49.56	12.52	4.805	3.542	-	-	0.574
Grount Truth (Waltz)	-	36.14	20.24	5.377	3.389	-	-	0.544
Grount Truth (50%)	-	12.26	7.98	4.718	3.665	-	-	0.688
Grount Truth (100%)	**1921**	6.05	6.839	4.867	3.598	-	-	0.6377
Grount Truth (Aug)	-	5.48	6.179	4.897	3.545	-	-	0.6377
Methods								
DuoDance-20	738	56.04	38.2	**4.761**	2.921	4.5174	2.369	0.356
DuoDance-Waltz	935	66.17	21.13	5.220	3.111	4.641	2.481	0.342
DuoDance-50	1411	**46.37**	15.77	5.136	3.396	4.639	2.528	0.365
DuoDance-100	1434	51.30	15.71	4.599	3.307	4.677	**2.544**	0.262
DuoDance-Baseline	1536	49.47	15.55	4.718	3.493	**4.680**	2.536	0.37
DuoDance-Mel	1555	49.45	**14.88**	4.966	3.352	4.612	2.529	**0.25**
DuoDance-Aug	1111	56.184	16.055	4.704	**3.443**	4.595	2.507	0.320

Table 2: Comparison of Elo, FID, Diversity, Multimodality, and Physical Foot Contact Metrics. We report the FID and Div metrics for each subset of the ground truth data on the test set against the training set. Please refer to Section 4.1.3 to see which dataset is used for each model. Note: ↑ means higher is better, → means closer to ground truth is better, and ↓ means lower is better.

5.2 Analysis of Data Availability

Our initial assumption was that we would see a strong correlation between the amount of data used to train DuoDance and the overall performance, both quantitative and qualitative. Our results indicate that this is not necessarily the case. Despite having the closest Div$_k$ to ground truth data DuoDance-20 is our worst-performing model qualitatively and scores high values for FID$_g$ and Div$_g$. DuoDance-50 scores lowest for FID$_k$ and is shown to perform close to DuoDance-100 qualitatively. With the introduction of augmented data for DuoDance-Aug, we see an increase in FID$_k$ and FID$_g$ and a reduction in user preference. While there is no correlation between Elo and FID$_k$, we do see a correlation between FID$_g$ and the user rankings. Models with a lower FID for geometric features scored higher in the user survey. There is little difference The poor performance of DuoDance-Aug may be related to convergence times. All models were trained for 1500 epochs. DuoDance-Aug is trained on twice the amount of data as DuoDance-100 and four times more than DuoDance-50. It is possible, that DuoDance-Aug would benefit from a longer training time. Both InterGen and EDGE are trained for 2000 epochs, and Tseng et al. [5] show that their fully trained model performs better qualitatively than all previous checkpoints. We do not believe this is a result of swapping the left and right sides of the motions as this same data augmentation technique is applied to the InterHuman dataset for InterGen training [19].

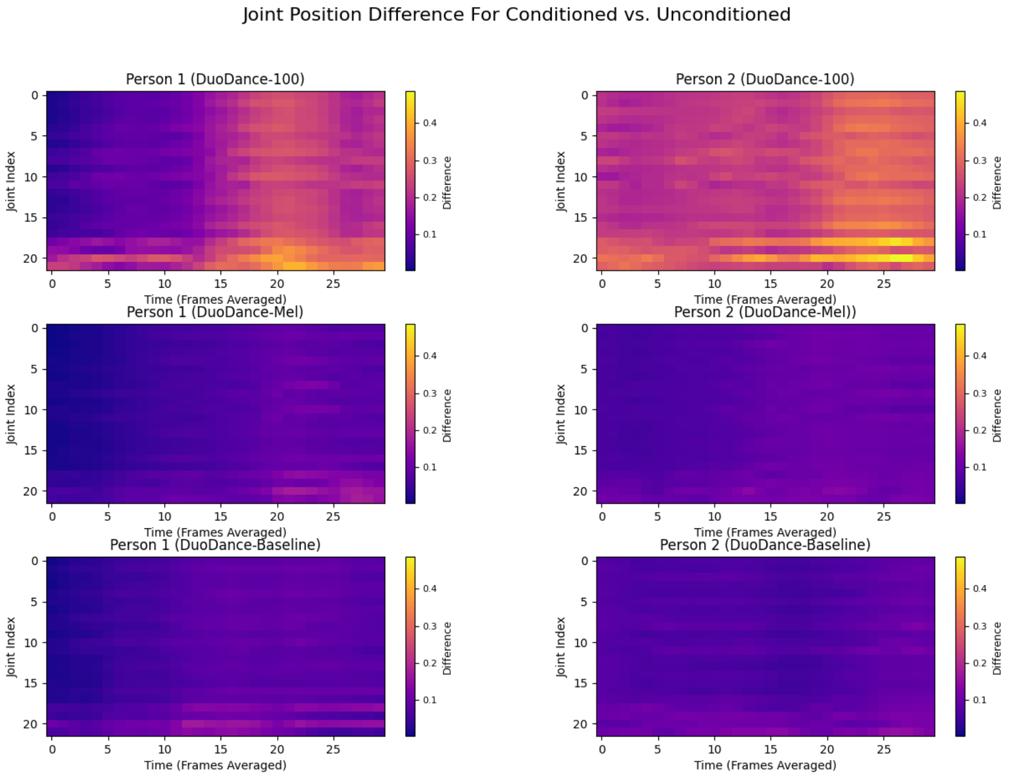


Figure 13: Heatmaps showing the average joint position difference per person for conditioned and unconditioned motion using the same Gaussian noise distribution. Results from top to bottom: DuoDance-100, DuoDance-Mel, DuoDance-Baseline. Difference is measured in meters.

5.3 Audio Representation Comparisons

Quantitative analysis shows all three audio representations perform similarly. Of the three variants trained on 100% of the data, DuoDance Mel scores lowest on FID_k and FID_g while having the closest Div_k to the ground truth. DuoDance-Baseline has the highest MM_k score. DuoDance-100 performs the worst on FID_k, FID_g, Div_k, and Div_g and only outperforms the other models on MM_g. Despite DuoDance-100 scoring lower than DuoDance-Baseline and DuoDance-Mel on the user survey, it was preferred over the ground-truth motion by 40% of participants. These results indicate a need for more thorough user evaluation where hundreds of samples are generated between each model pair instead of only one. We find that our current architecture learns a stronger conditioning signal from Jukebox features than from Mel and Baseline. To demonstrate this, we generate two motions for each model. One motion is conditioned on the audio, while the other motion has the 100% of the audio embeddings set to $\emptyset$. Both models use the same Gaussian noise distribution. We transform the motion such that person 1 is at the origin and facing the $+z$-axis. Person 2's motion is then transformed accordingly to retain the relative positional and rotational information. We then compute the average joint distance per person between the conditioned and unconditioned motion averaged every five frames and display the results as a heatmap (see Figure 13). We can see that DuoDance-100 has the most variation between conditioned and unconditioned motion when focusing on joint positions. DuoDance-Mel and DuoDance-Baseline, have slight variations but their overall motion remains very similar (see Figure 14). Despite the higher variance in joint positions, DuoDance-100 does not consistently produce highly different motion, as can be seen in Figure 15.

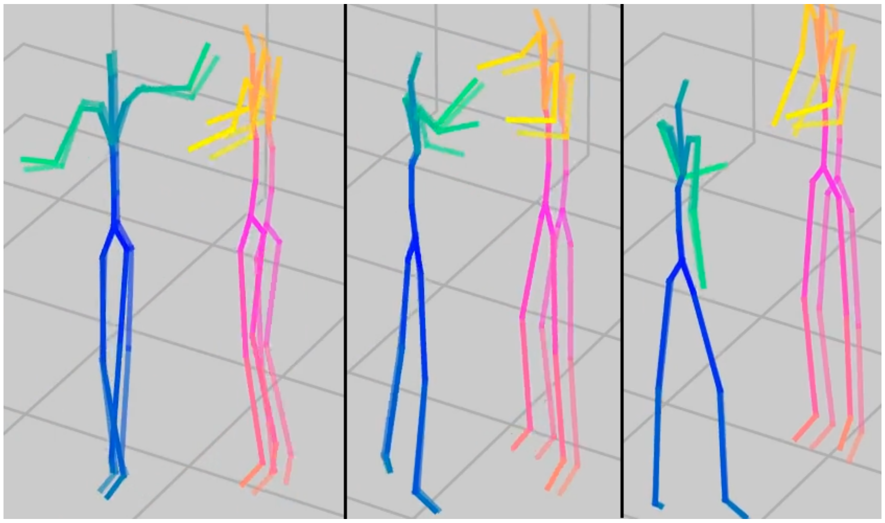


Figure 14: Three frames taken from generated motion of DuoDance-Mel. Both conditioned and unconditioned motions are layered atop each other, with the unconditioned motion displayed slightly transparent.

5.4 User Survey

In total, 10 survey responses were collected. These surveys were distributed online to willing volunteers who received no remuneration. Participants varied in age as shown in following age distribution: 30% in the 18-25 range, 40% in the 26-35 range, 10% in the 36-45 range, and 20% over 55. 60% of the participants identify as male while 30% identify as male, and 10% identify as non-binary/other. Before evaluating the generated motion, users are asked to rate their familiarity with partner dancing as well as 3D animation videos. We compute an average familiarity value of 2.8 for partner dancing and 3.4 for 3D animated videos on a 5-point likert scale. We show the users 28 pairs of videos of generated motion paired with audio, each video 10 seconds long. Both videos within a pair use a different model to generate motion from the same audio. The order in which the videos are shown is randomized to avoid the impact of random selection. The resulting win table can be seen in Table 3 which is then used to compute the overall Elo ranking of each model. The user-preferred motion was considered more complex and variable in 87% of the comparisons, more physically plausible in 79% of the comparisons, and better suited for the music in 84% of the comparisons. This may indicate a slight preference for motion complexity over realism and suitability for audio. This is further exemplified by comparing the results of DuoDance-100 and the ground truth data. Both were considered to be equally suitable for music and physically plausible, however, 70% of participants found the ground truth to be more complex resulting in a 60% overall preference.

We only had the capacity to generate one comparison each pair. This can significantly alter the qualitative results as a single sample is not indicative of the model's overall performance. Ideally, we would generate upwards of 100 samples for each pair-wise comparison. These samples would be added to different surveys and we would collect a more representative ranking distribution.

5.5 Cross-Attention Impact

One of the primary design choices of DuoDance is the addition of FiLM layers and cross-attention layer for the audio between the self and mutual-attention mechanisms. This differs from InterGen, which uses adaptive layer normalization to condition on the text condition and timestep. If we exclude the cross-attention and FiLM layers, and use adaptive layer normalization for DuoDance, we find the model is incapable of learning from the conditioning. Instead, it learns from only the Gaussian noise distribution to generate its results. When using our novel architecture, we see the conditioning strength is increased for generating motion. To demonstrate this impact, we perform a similar comparison between joint positions of conditioned and unconditioned motion as in Section 5.3. The resulting heatmap can be seen in Figure 17.

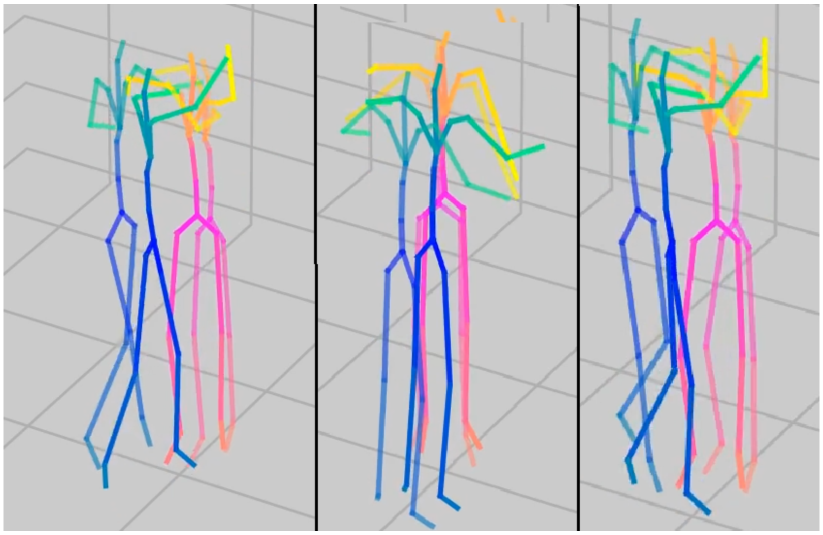


Figure 15: Three frames taken from generated motion of DuoDance-100. Both conditioned and unconditioned motions are layered atop each other, with the unconditioned motion displayed slightly transparent.

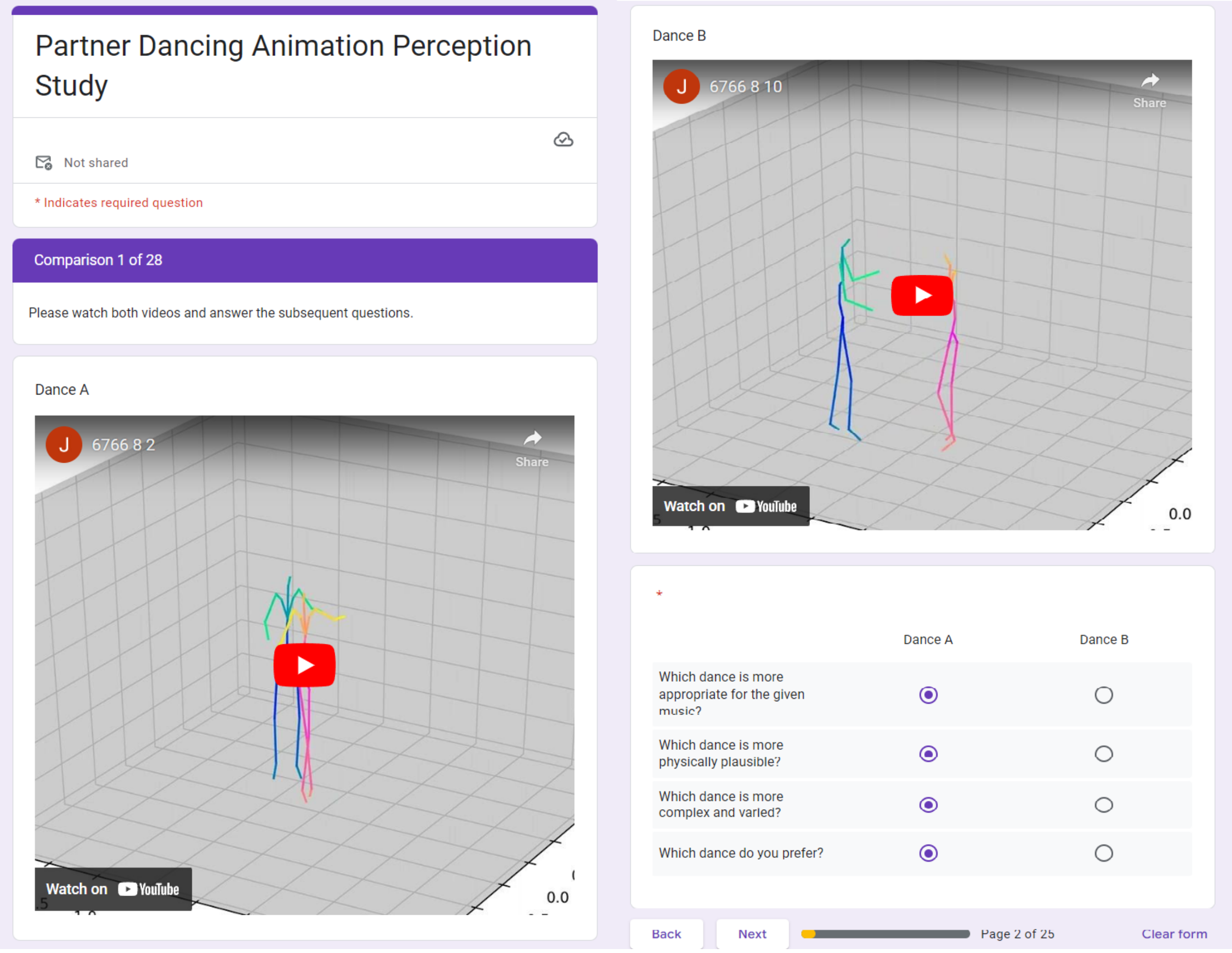
Model A / Model B	20	W	50	100	B	M	A	GT
20	0	0	0	1	1	0	0	1
W	10	0	2	2	2	3	3	0
50	10	8	0	6	0	2	8	2
100	9	8	4	0	2	9	9	4
B	9	8	10	8	0	0	8	3
M	10	7	8	1	10	0	10	1
A	10	7	2	1	2	0	0	1
GT	9	10	8	6	7	9	9	0

Table 3: This table shows the number of wins of Model A over Model B. For model names, 20 refers to DuoDance-20, W refers to DuoDance-Waltz, 50 refers to DuoDance-50, 100 refers to DuoDance-100, M refers to DuoDance-Mel, B refers to DuoDance-Baseline, A refers to DuoDance-Aug, and GT refers to the ground truth motion.

6 Limitations and Future Work

When comparing our artificially constructed dataset to other datasets used in SOTA dance generation, there are stark differences. The first is that we do not know with certainty what song was danced on. Enlisting the help of a dance expert to learn the style of dance helped narrow down possible audio choices, however, we may be missing crucial audio-motion mappings or inadvertently establishing false ones. Audio selection is also a difficult task for ballroom dancing as modern pop songs are sometimes used for Waltz, Jive, Cha-Cha. Therefore, the same dance can be performed on greatly different music. Ideally, we would focus on two or three specific styles of dance that are characterised strongly by their music, e.g. Salsa, Bachata, Kizomba. By pairing the same motion with multiple songs, we may make it more difficult for our model to learn relevant mappings between the audio and motion features. Many datasets have their dancers perform the same choreography on different music, but each performance is unique and may match slow-down/speed-up in order to match the audio. This creates more variance than simply reusing the exact same motion data. Finally, our dataset is relatively small when compared to datasets such as AIST++ [103] which has over 1400 unique sequences totaling over 5 hours.

Our comparisons detailing the impact of conditioning only use joint positioning information. It is possible to have two identical sets of motion that are translated along the horizontal and vertical planes. In this instance, our comparison would show high variability in motion, despite joint rotational information being the same.



Partner Dancing Animation Perception Study

Not shared

* Indicates required question

Comparison 1 of 28

Please watch both videos and answer the subsequent questions.

Dance A

Dance B

*

	Dance A	Dance B
Which dance is more appropriate for the given music?	●	○
Which dance is more physically plausible?	●	○
Which dance is more complex and varied?	●	○
Which dance do you prefer?	●	○

Back Next Page 2 of 25 Clear form

Figure 16: Example screenshots of the user survey. Note: Dance B is shown below Dance A on the form; however, for the purpose of this report they are shown side-by-side. Survey wording and formatting taken from [102].

Ideally, a better comparison would be made which reconstructs the joint angles using inverse kinematics and compares these in addition to the root joint to gain a more thorough representation of the motion.
Our user survey only consisted of 10 participants and 56 clips, far from the 147 raters and 11,610 clips used to evaluate EDGE. Considering motions are only paired once for comparison, it is impossible to justify claims that one model outperforms another for all music genres and dance styles. Constructing thousands of samples across a wide variety of music.
Denoising Diffusion Implicit Models [67] have been shown to produce high-quality samples 10x to 50x faster than traditional denoising diffusion probabilistic models. While it is commonly used in diffusion-based motion generation [5, 19], there have been no investigations into the impact of sample quality for motion generation. Based on an initial visual inspection of generated motion using the same conditioning and Gaussian noise distribution, we see a large impact on the resulting motion depending on the number of sample timesteps as seen in Figure 18. It is possible that more sampling steps result in significantly higher-quality samples for motion generation.

7 Conclusion

Our first research question was the following:

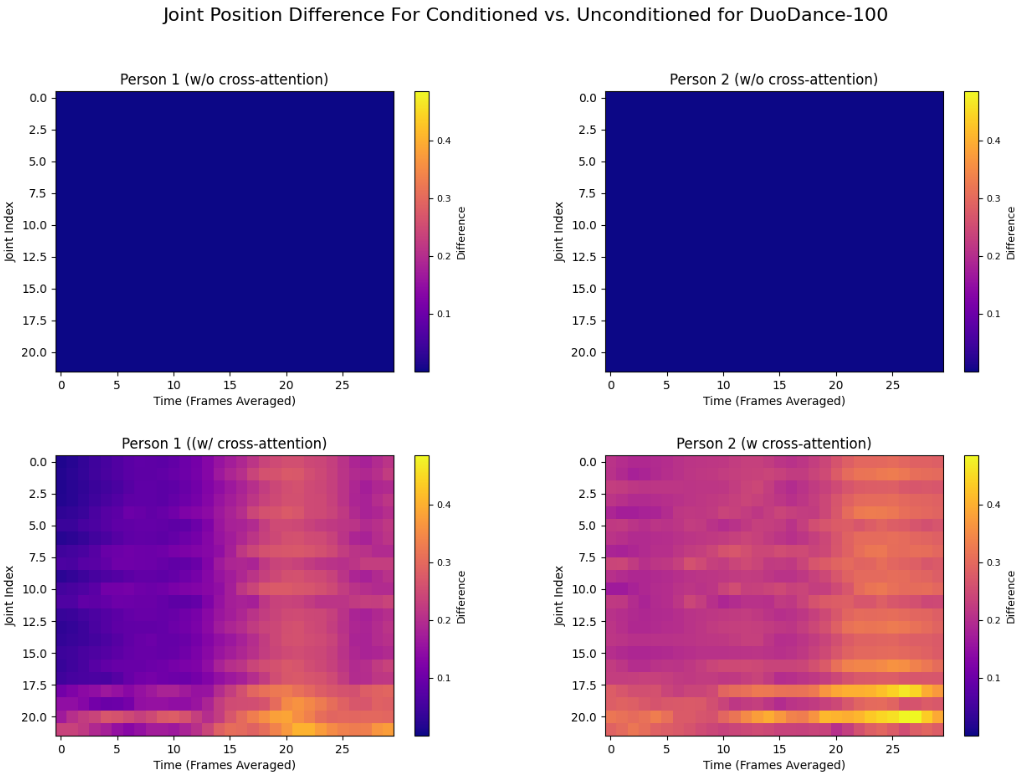


Figure 17: Heatmaps showing the average joint position difference per person for conditioned and unconditioned motion using the same Gaussian noise distribution. Difference is measured in meters.

- **How does the inclusion of different audio features impact the results?**

To answer this question, we compared three models each using a different audio representation: “Baseline”, Mel-Spectrogram, and Jukebox features. Initial quantitative and qualitative metrics show a preference for motion generated using Mel-Spectrogram features, however, a deeper analysis shows our model cannot learn strong correlations between Mel and “Baseline” features and the motion features. Instead, our model generates motion primarily influenced by noise which fits the overall motion distribution.
Our second research question asked:

- **What representative features of interactive dance motion are most suitable for quantitative evaluation?**

Unfortunately, we were only able to determine which features are not representative for quantitative evaluation. We show that a lower FID_k score does not lead to higher-quality motion which is inline with the findings of Tseng et al. [5]. We do see a correlation between user preference and FID_g, but given the small sample size we cannot deem this as a significant result. We also show the limitations of the Physical Foot Contact score and conclude the metric in its current definition is an unreliable approach to measuring physical plausibility for two-person motion generation.
Our final research equation:

- **How does the amount of data used to train our model affect the resulting motion?**

We find that all of our models outperformed DuoDance-20 which was trained on the least amount of data. Survey results showed that our model trained on 50% and 100% of the data scored similarly. Our model trained on Augmented data performed the worst, but this may be due to a lack of convergence given the same number of Epochs for 2x the amount of the data.

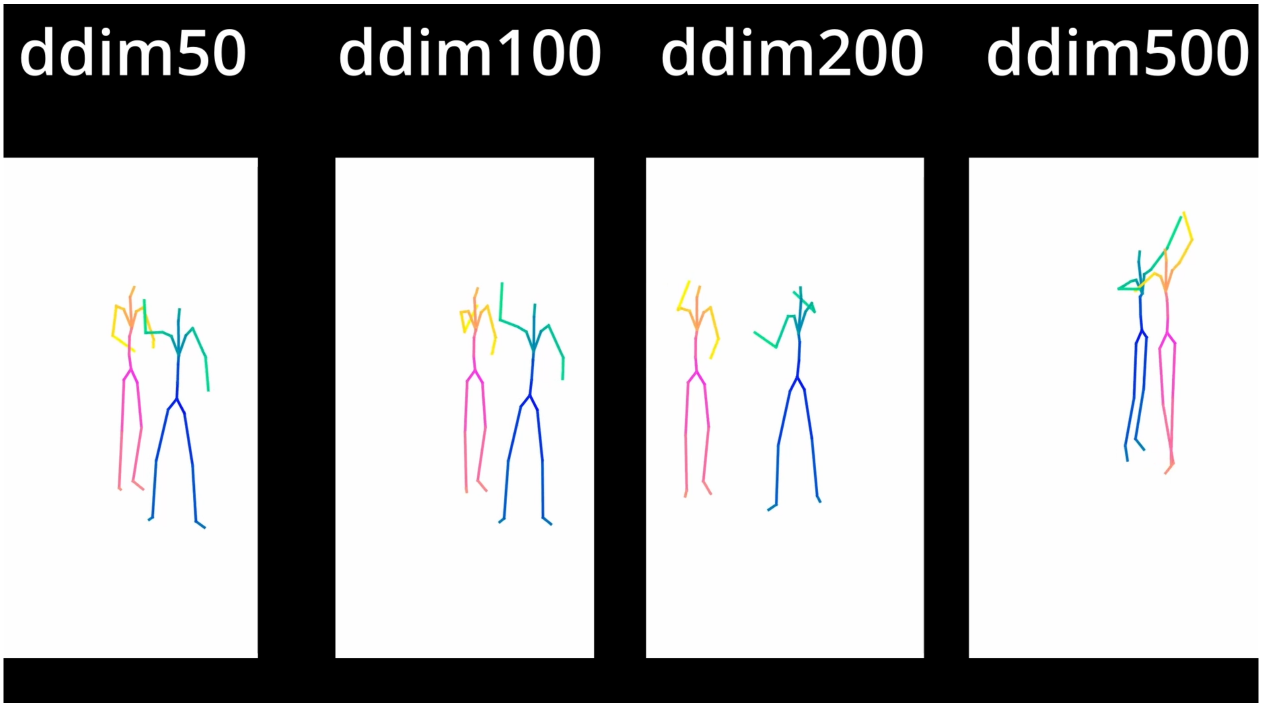


Figure 18: Motion snapshots taken at a variable time for the same conditioning input and Gaussian noise distribution using different sampling timesteps. Using 50 and 100 timesteps results in similar motion, but looking at the results from 200 and 500 shows that the quality of the sample may change drastically unlike in the image domain.

In conclusion, we adapt the mutual-attention architecture of InterGen to generate 3D motion of partner dancing conditioned on music, the first diffusion-based approach to do so. By incorporating a cross-attention layer between self-attention and mutual-attention layers, we strengthen the conditioning signal of the input audio. Qualitative and quantitative analysis shows that while our method can produce motion more favorable than the ground truth, the strength of the conditioning should be further improved to generate more diverse motion.

References

[1] Tae-hoon Kim, Sang Il Park, and Sung Yong Shin. “Rhythmic-Motion Synthesis Based on Motion-Beat Analysis”. In: *ACM Trans. Graph.* 22.3 (July 2003), pp. 392–401. ISSN: 0730-0301. DOI: 10.1145/882262.882283. URL: https://doi.org/10.1145/882262.882283.

[2] Buyu Li, Yongchi Zhao, and Lu Sheng. “DanceNet3D: Music Based Dance Generation with Parametric Motion Transformer”. In: *CoRR* abs/2103.10206 (2021). arXiv: 2103.10206. URL: https://arxiv.org/abs/2103.10206.

[3] Ruilong Li et al. “Learn to Dance with AIST++: Music Conditioned 3D Dance Generation”. In: *CoRR* abs/2101.08779 (2021). arXiv: 2101.08779. URL: https://arxiv.org/abs/2101.08779.

[4] Jiaman Li et al. “Learning to Generate Diverse Dance Motions with Transformer”. In: *CoRR* abs/2008.08171 (2020). arXiv: 2008.08171. URL: https://arxiv.org/abs/2008.08171.

[5] Jonathan Tseng, Rodrigo Castellon, and C. Karen Liu. *EDGE: Editable Dance Generation From Music*. 2022. arXiv: 2211.10658 [cs.SD].

[6] Simon Alexanderson et al. *Listen, denoise, action! Audio-driven motion synthesis with diffusion models*. 2022. arXiv: 2211.09707 [cs.LG].

[7] Andreas Aristidou et al. “Rhythm is a Dancer: Music-Driven Motion Synthesis with Global Structure”. In: *CoRR* abs/2111.12159 (2021). arXiv: 2111.12159. URL: https://arxiv.org/abs/2111.12159.

[8] Rishabh Dabral et al. *MoFusion: A Framework for Denoising-Diffusion-based Motion Synthesis*. 2022. arXiv: 2212.04495 [cs.CV].

[9] Takaaki Shiratori, Atsushi Nakazawa, and Katsushi Ikeuchi. “Dancing-to-Music Character Animation”. In: *Comput. Graph. Forum* 25 (Sept. 2006), pp. 449–458. DOI: 10.1111/j.1467-8659.2006.00964.x.

[10] Danielle Sauer and Yee-Hong Yang. “Music-Driven Character Animation”. In: 5.4 (Nov. 2009). ISSN: 1551-6857. DOI: 10.1145/1596990.1596991. URL: https://doi.org/10.1145/1596990.1596991.

[11] Omid Alemi, Jules Françoise, and Philippe Pasquier. “GrooveNet: Real-Time Music-Driven Dance Movement Generation using Artificial Neural Networks”. In: Aug. 2017.

[12] Taoran Tang, Jia Jia, and Hanyang Mao. “Dance with Melody: An LSTM-Autoencoder Approach to Music-Oriented Dance Synthesis”. In: *Proceedings of the 26th ACM International Conference on Multimedia*. MM ’18. Seoul, Republic of Korea: Association for Computing Machinery, 2018, pp. 1598–1606. ISBN: 9781450356657. DOI: 10.1145/3240508.3240526. URL: https://doi.org/10.1145/3240508.3240526.

[13] Yu Qi, Yazhou Liu, and Quansen Sun. “Music-Driven Dance Generation”. In: *IEEE Access* 7 (2019), pp. 166540–166550. DOI: 10.1109/ACCESS.2019.2953698.

[14] Hsin-Ying Lee et al. “Dancing to Music”. In: *Neural Information Processing Systems*. 2019.

[15] Guofei Sun et al. “DeepDance: Music-to-Dance Motion Choreography With Adversarial Learning”. In: *IEEE Transactions on Multimedia* 23 (2021), pp. 497–509. DOI: 10.1109/TMM.2020.2981989.

[16] Ruozi Huang et al. “Dance Revolution: Long Sequence Dance Generation with Music via Curriculum Learning”. In: *CoRR* abs/2006.06119 (2020). arXiv: 2006.06119. URL: https://arxiv.org/abs/2006.06119.

[17] Nhat Le et al. *Music-Driven Group Choreography*. 2023. arXiv: 2303.12337 [cs.MM].

[18] Zixuan Wang et al. “GroupDancer: Music to Multi-People Dance Synthesis with Style Collaboration”. In: *Proceedings of the 30th ACM International Conference on Multimedia*. MM ’22. Lisboa, Portugal: Association for Computing Machinery, 2022, pp. 1138–1146. ISBN: 9781450392037. DOI: 10.1145/3503161.3548090. URL: https://doi.org/10.1145/3503161.3548090.

[19] Han Liang et al. *InterGen: Diffusion-based Multi-human Motion Generation under Complex Interactions*. 2023. arXiv: 2304.05684 [cs.CV].

[20] Yonatan Shafir et al. *Human Motion Diffusion as a Generative Prior*. 2023. arXiv: 2303.01418 [cs.CV].

[21] H. Van Welbergen et al. "Real Time Animation of Virtual Humans: A Trade-off Between Naturalness and Control". In: *Computer Graphics Forum* 29.8 (2010), pp. 2530–2554. DOI: https://doi.org/10.1111/j.1467-8659.2010.01822.x. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1467-8659.2010.01822.x. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-8659.2010.01822.x.

[22] L. Mourot et al. "A Survey on Deep Learning for Skeleton-Based Human Animation". In: *Computer Graphics Forum* 41.1 (Feb. 2022), pp. 122–157. ISSN: 0167-7055, 1467-8659. DOI: 10.1111/cgf.14426. arXiv: 2110.06901 [cs]. (Visited on 03/12/2023).

[23] Dario Pavllo, David Grangier, and Michael Auli. *QuaterNet: A Quaternion-Based Recurrent Model for Human Motion*. July 2018. arXiv: arXiv:1805.06485. (Visited on 03/14/2023).

[24] Daniel Holden et al. "Learning motion manifolds with convolutional autoencoders". en. In: *SIGGRAPH Asia 2015 Technical Briefs*. Kobe Japan: ACM, Nov. 2015, pp. 1–4. ISBN: 978-1-4503-3930-8. DOI: 10.1145/2820903.2820918. URL: https://dl.acm.org/doi/10.1145/2820903.2820918 (visited on 03/15/2023).

[25] Daniel Holden, Jun Saito, and Taku Komura. "A Deep Learning Framework for Character Motion Synthesis and Editing". In: *ACM Transactions on Graphics* 35.4 (July 2016), pp. 1–11. ISSN: 0730-0301, 1557-7368. DOI: 10.1145/2897824.2925975.

[26] Mingyi Shi et al. *MotioNet: 3D Human Motion Reconstruction from Monocular Video with Skeleton Consistency*. June 2020. arXiv: arXiv:2006.12075.

[27] Nefeli Andreou, Andreas Aristidou, and Yiorgos Chrysanthou. *Pose Representations for Deep Skeletal Animation*. July 2022. eprint: arXiv:2111.13907.

[28] Gustav Eje Henter, Simon Alexanderson, and Jonas Beskow. "MoGlow: Probabilistic and controllable motion synthesis using normalising flows". In: *CoRR* abs/1905.06598 (2019). arXiv: 1905.06598. URL: http://arxiv.org/abs/1905.06598.

[29] Julieta Martinez, Michael J. Black, and Javier Romero. "On human motion prediction using recurrent neural networks". In: *CoRR* abs/1705.02445 (2017). arXiv: 1705.02445. URL: http://arxiv.org/abs/1705.02445.

[30] Sigal Raab et al. "MoDi: Unconditional Motion Synthesis from Diverse Data". In: *ArXiv* abs/2206.08010 (2022).

[31] Yi Zhou et al. *On the Continuity of Rotation Representations in Neural Networks*. 2018. DOI: 10.48550/ARXIV.1812.07035. URL: https://arxiv.org/abs/1812.07035.

[32] Chuan Guo et al. "Generating Diverse and Natural 3D Human Motions From Text". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2022, pp. 5152–5161.

[33] Jihoon Kim, Jiseob Kim, and Sungjoon Choi. *FLAME: Free-form Language-based Motion Synthesis & Editing*. 2022. DOI: 10.48550/ARXIV.2209.00349. URL: https://arxiv.org/abs/2209.00349.

[34] Jianxin Ma, Shuai Bai, and Chang Zhou. "Pretrained Diffusion Models for Unified Human Motion Synthesis". In: *ArXiv* abs/2212.02837 (2022).

[35] Mingyuan Zhang et al. *MotionDiffuse: Text-Driven Human Motion Generation with Diffusion Model*. 2022. arXiv: 2208.15001 [cs.CV].

[36] Sigal Raab et al. *Single Motion Diffusion*. 2023. arXiv: 2302.05905 [cs.CV].

[37] Kfir Aberman et al. "Skeleton-Aware Networks for Deep Motion Retargeting". In: *ACM Transactions on Graphics (TOG)* 39.4 (2020), p. 62.

[38] Kyungho Lee, Seyoung Lee, and J. Lee. "Interactive character animation by learning multi-objective control". In: vol. 37. Dec. 2018, pp. 1–10. DOI: 10.1145/3272127.3275071.

[39] Sebastian Starke et al. "Local Motion Phases for Learning Multi-Contact Character Movements". In: 39.4 (Aug. 2020). ISSN: 0730-0301. DOI: 10.1145/3386569.3392450. URL: https://doi.org/10.1145/3386569.3392450.

[40] He Zhang et al. "Mode-Adaptive Neural Networks for Quadruped Motion Control". In: 37.4 (July 2018). ISSN: 0730-0301. DOI: 10.1145/3197517.3201366. URL: https://doi.org/10.1145/3197517.3201366.

[41] Hung Yu Ling et al. "Character Controllers Using Motion VAEs". In: *CoRR* abs/2103.14274 (2021). arXiv: 2103.14274. URL: https://arxiv.org/abs/2103.14274.

[42] Daniel Holden et al. "Learned Motion Matching". In: 39.4 (Aug. 2020). ISSN: 0730-0301. DOI: 10.1145/3386569.3392440. URL: https://doi.org/10.1145/3386569.3392440.

[43] Nefeli Andreou et al. "A Hierarchy-Aware Pose Representation for Deep Character Animation". In: *CoRR* abs/2111.13907 (2021). arXiv: 2111.13907. URL: https://arxiv.org/abs/2111.13907.

[44] Kedi Lyu et al. *3D Human Motion Prediction: A Survey.* 2022. arXiv: 2203.01593 [cs.CV].

[45] Richard Bowden and Ub Ph. "Learning Statistical Models of Human Motion". In: (Oct. 2000).

[46] Vladimir Pavlovic et al. "A Dynamic Bayesian Network Approach to Figure Tracking Using Learned Dynamic Models". In: *Proceedings of the IEEE International Conference on Computer Vision* 1 (May 2001).

[47] Jack M. Wang, David J. Fleet, and Aaron Hertzmann. "Gaussian Process Dynamical Models for Human Motion". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 30.2 (2008), pp. 283–298. DOI: 10.1109/TPAMI.2007.1167.

[48] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. "Sequence to Sequence Learning with Neural Networks". In: *CoRR* abs/1409.3215 (2014). arXiv: 1409.3215. URL: http://arxiv.org/abs/1409.3215.

[49] Katerina Fragkiadaki, Sergey Levine, and Jitendra Malik. "Recurrent Network Models for Kinematic Tracking". In: *CoRR* abs/1508.00271 (2015). arXiv: 1508.00271. URL: http://arxiv.org/abs/1508.00271.

[50] Liang-Yan Gui et al. "Adversarial Geometry-Aware Human Motion Prediction". In: *Computer Vision – ECCV 2018*. Ed. by Vittorio Ferrari et al. Cham: Springer International Publishing, 2018. ISBN: 978-3-030-01225-0.

[51] Wei Mao et al. "Learning Trajectory Dependencies for Human Motion Prediction". In: *CoRR* abs/1908.05436 (2019). arXiv: 1908.05436. URL: http://arxiv.org/abs/1908.05436.

[52] Wei Mao, Miaomiao Liu, and Mathieu Salzmann. "History Repeats Itself: Human Motion Prediction via Motion Attention". In: *CoRR* abs/2007.11755 (2020). arXiv: 2007.11755. URL: https://arxiv.org/abs/2007.11755.

[53] Lingwei Dang et al. "MSR-GCN: Multi-Scale Residual Graph Convolution Networks for Human Motion Prediction". In: *CoRR* abs/2108.07152 (2021). arXiv: 2108.07152. URL: https://arxiv.org/abs/2108.07152.

[54] Maosen Li et al. "Dynamic Multiscale Graph Neural Networks for 3D Skeleton-Based Human Motion Prediction". In: *CoRR* abs/2003.08802 (2020). arXiv: 2003.08802. URL: https://arxiv.org/abs/2003.08802.

[55] Maosen Li et al. "Symbiotic Graph Neural Networks for 3D Skeleton-based Human Action Recognition and Motion Prediction". In: *CoRR* abs/1910.02212 (2019). arXiv: 1910.02212. URL: http://arxiv.org/abs/1910.02212.

[56] Emad Barsoum, John Kender, and Zicheng Liu. "HP-GAN: Probabilistic 3D human motion prediction via GAN". In: *CoRR* abs/1711.09561 (2017). arXiv: 1711.09561. URL: http://arxiv.org/abs/1711.09561.

[57] Jogendra Nath Kundu, Maharshi Gor, and R. Venkatesh Babu. "BiHMP-GAN: Bidirectional 3D Human Motion Prediction GAN". In: *CoRR* abs/1812.02591 (2018). arXiv: 1812.02591. URL: http://arxiv.org/abs/1812.02591.

[58] Qiongjie Cui et al. "Efficient human motion prediction using temporal convolutional generative adversarial network". In: *Information Sciences* 545 (2021), pp. 427–447. ISSN: 0020-0255. DOI: https://doi.org/10.1016/j.ins.2020.08.123. URL: https://www.sciencedirect.com/science/article/pii/S0020025520308896.

[59] Zhenguang Liu et al. “Aggregated Multi-GANs for Controlled 3D Human Motion Prediction”. In: *CoRR* abs/2103.09755 (2021). arXiv: 2103.09755. URL: https://arxiv.org/abs/2103.09755.

[60] Ikhsanul Habibie et al. “A Recurrent Variational Autoencoder for Human Motion Synthesis”. In: *British Machine Vision Conference*. 2017.

[61] Chuan Guo et al. “Action2Motion: Conditioned Generation of 3D Human Motions”. In: *CoRR* abs/2007.15240 (2020). arXiv: 2007.15240. URL: https://arxiv.org/abs/2007.15240.

[62] Mathis Petrovich, Michael J. Black, and Gül Varol. “Action-Conditioned 3D Human Motion Synthesis with Transformer VAE”. In: *CoRR* abs/2104.05670 (2021). arXiv: 2104.05670. URL: https://arxiv.org/abs/2104.05670.

[63] Haoye Cai et al. “Deep Video Generation, Prediction and Completion of Human Action Sequences”. In: *CoRR* abs/1711.08682 (2017). arXiv: 1711.08682. URL: http://arxiv.org/abs/1711.08682.

[64] S. Yan et al. “Convolutional Sequence Generation for Skeleton-Based Action Synthesis”. In: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. Los Alamitos, CA, USA: IEEE Computer Society, Nov. 2019, pp. 4393–4401. DOI: 10.1109/ICCV.2019.00449. URL: https://doi-ieeecomputersociety-org.proxy.library.uu.nl/10.1109/ICCV.2019.00449.

[65] Bruno Degardin et al. “Generative Adversarial Graph Convolutional Networks for Human Action Synthesis”. In: *CoRR* abs/2110.11191 (2021). arXiv: 2110.11191. URL: https://arxiv.org/abs/2110.11191.

[66] Jascha Sohl-Dickstein et al. “Deep Unsupervised Learning using Nonequilibrium Thermodynamics”. In: *CoRR* abs/1503.03585 (2015). arXiv: 1503.03585. URL: http://arxiv.org/abs/1503.03585.

[67] Jiaming Song, Chenlin Meng, and Stefano Ermon. “Denoising Diffusion Implicit Models”. In: *CoRR* abs/2010.02502 (2020). arXiv: 2010.02502. URL: https://arxiv.org/abs/2010.02502.

[68] Jonathan Ho, Ajay Jain, and Pieter Abbeel. “Denoising Diffusion Probabilistic Models”. In: *CoRR* abs/2006.11239 (2020). arXiv: 2006.11239. URL: https://arxiv.org/abs/2006.11239.

[69] Prafulla Dhariwal and Alex Nichol. “Diffusion Models Beat GANs on Image Synthesis”. In: *CoRR* abs/2105.05233 (2021). arXiv: 2105.05233. URL: https://arxiv.org/abs/2105.05233.

[70] Guy Tevet et al. *Human Motion Diffusion Model*. 2022. arXiv: 2209.14916 [cs.CV].

[71] Mingyuan Zhang et al. *ReMoDiffuse: Retrieval-Augmented Motion Diffusion Model*. 2023. arXiv: 2304.01116 [cs.CV].

[72] M.R. Reshma et al. “Cultural heritage preservation through dance digitization: A review”. In: *Digital Applications in Archaeology and Cultural Heritage* 28 (2023), e00257. ISSN: 2212-0548. DOI: https://doi.org/10.1016/j.daach.2023.e00257. URL: https://www.sciencedirect.com/science/article/pii/S2212054823000024.

[73] Simon Senecal et al. “Salsa dance learning evaluation and motion analysis in gamified virtual reality environment”. In: *Multimedia Tools and Applications* 79 (Sept. 2020). DOI: 10.1007/s11042-020-09192-y.

[74] Luka Crnkovic-Friis and Louise Crnkovic-Friis. “Generative Choreography using Deep Learning”. In: *CoRR* abs/1605.06921 (2016). arXiv: 1605.06921. URL: http://arxiv.org/abs/1605.06921.

[75] Zhifeng Kong et al. *DiffWave: A Versatile Diffusion Model for Audio Synthesis*. 2021. arXiv: 2009.09761 [eess.AS].

[76] Prafulla Dhariwal et al. *Jukebox: A Generative Model for Music*. 2020. arXiv: 2005.00341 [eess.AS].

[77] Javier Pérez-Marcos et al. “Multi-agent system application for music features extraction, meta-classification and context analysis”. In: *Knowledge and Information Systems* 62 (Jan. 2020). DOI: 10.1007/s10115-018-1319-2.

[78] Geoffroy Peeters. “A large set of audio features for sound description (similarity and classification) in the CUIDADO project”. In: (Jan. 2004).

[79] Yang Zhou et al. “VisemeNet: Audio-Driven Animator-Centric Speech Animation”. In: *CoRR* abs/1805.09488 (2018). arXiv: 1805.09488. URL: http://arxiv.org/abs/1805.09488.

[80] Xin Wen et al. "Photorealistic Audio-driven Video Portraits". In: *IEEE Transactions on Visualization and Computer Graphics* 26.12 (2020), pp. 3457–3466. DOI: 10.1109/TVCG.2020.3023573.

[81] Alysha Bogaers, Zerrin Yumak, and Anja Volk. "Music-Driven Animation Generation of Expressive Musical Gestures". In: *Companion Publication of the 2020 International Conference on Multimodal Interaction*. ICMI '20 Companion. Virtual Event, Netherlands: Association for Computing Machinery, 2021, pp. 22–26. ISBN: 9781450380027. DOI: 10.1145/3395035.3425244. URL: https://doi.org/10.1145/3395035.3425244.

[82] Lian Siyao et al. "Bailando: 3D Dance Generation by Actor-Critic GPT with Choreographic Memory". In: *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2022), pp. 11040–11049.

[83] Rodrigo Castellon, Chris Donahue, and Percy Liang. "Codified audio language modeling learns useful representations for music information retrieval". In: *CoRR* abs/2107.05677 (2021). arXiv: 2107.05677. URL: https://arxiv.org/abs/2107.05677.

[84] Martin Heusel et al. "GANs Trained by a Two Time-Scale Update Rule Converge to a Nash Equilibrium". In: *CoRR* abs/1706.08500 (2017). arXiv: 1706.08500. URL: http://arxiv.org/abs/1706.08500.

[85] Kensuke Onuma, Christos Faloutsos, and Jessica Hodgins. "FMDistance: A fast and effective distance function for motion capture data". In: (Jan. 2008).

[86] Meinard Müller, Tido Röder, and Michael Clausen. "Efficient Content-Based Retrieval of Motion Capture Data". In: *ACM Trans. Graph.* 24.3 (July 2005), pp. 677–685. ISSN: 0730-0301. DOI: 10.1145/1073204.1073247. URL: https://doi.org/10.1145/1073204.1073247.

[87] Murchana Baruah and Bonny Banerjee. "A Multimodal Predictive Agent Model for Human Interaction Generation". In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 2020, pp. 4402–4410. DOI: 10.1109/CVPRW50498.2020.00519.

[88] Baptiste Chopin et al. *Interaction Transformer for Human Reaction Generation*. 2023. arXiv: 2207.01685 [cs.CV].

[89] Jogendra Nath Kundu et al. "Cross-Conditioned Recurrent Networks for Long-Term Synthesis of Inter-Person Human Motion Interactions". In: *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*. 2020, pp. 2713–2722. DOI: 10.1109/WACV45572.2020.9093627.

[90] Wen Guo et al. "Multi-Person Extreme Motion Prediction with Cross-Interaction Attention". In: *CoRR* abs/2105.08825 (2021). arXiv: 2105.08825. URL: https://arxiv.org/abs/2105.08825.

[91] Muhammad Rameez Ur Rahman et al. "Best Practices for 2-Body Pose Forecasting". In: *ArXiv* abs/2304.05758 (2023).

[92] URL: http://mocap.cs.cmu.edu/.

[93] Chitwan Saharia et al. *Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding*. arXiv:2205.11487 [cs]. May 2022. URL: http://arxiv.org/abs/2205.11487 (visited on 04/20/2024).

[94] Brian McFee et al. "librosa: Audio and music signal analysis in python". In: *Proceedings of the 14th python in science conference*. Vol. 8. 2015.

[95] Jonathan Shen et al. *Natural TTS Synthesis by Conditioning WaveNet on Mel Spectrogram Predictions*. arXiv:1712.05884 [cs]. Feb. 2018. URL: http://arxiv.org/abs/1712.05884 (visited on 04/20/2024).

[96] Yang Song et al. *Score-Based Generative Modeling through Stochastic Differential Equations*. arXiv:2011.13456 [cs, stat]. Feb. 2021. URL: http://arxiv.org/abs/2011.13456.

[97] Ethan Perez et al. "FiLM: Visual Reasoning with a General Conditioning Layer". In: *CoRR* abs/1709.07871 (2017). arXiv: 1709.07871. URL: http://arxiv.org/abs/1709.07871.

[98] Alec Radford et al. *Learning Transferable Visual Models From Natural Language Supervision*. 2021. arXiv: 2103.00020 [cs.CV].

[99] Antoine Maiorca, Youngwoo Yoon, and Thierry Dutoit. *Evaluating the Quality of a Synthesized Motion with the Fréchet Motion Distance*. 2022. arXiv: 2204.12318 [cs.CV].

[100] Antoine Maiorca, Youngwoo Yoon, and Thierry Dutoit. “Validating Objective Evaluation Metric: Is Fréchet Motion Distance able to Capture Foot Skating Artifacts ?” In: *Proceedings of the 2023 ACM International Conference on Interactive Media Experiences* (2023). URL: https://api.semanticscholar.org/CorpusID:261279024.

[101] Ilya Loshchilov and Frank Hutter. *Decoupled Weight Decay Regularization.* 2019. arXiv: 1711.05101 [cs.LG].

[102] Stefan Stan, Kazi Injamamul Haque, and Zerrin Yumak. *FaceDiffuser: Speech-Driven 3D Facial Animation Synthesis Using Diffusion.* 2023. arXiv: 2309.11306 [cs.CV].

[103] Ruilong Li et al. “Learn to Dance with AIST++: Music Conditioned 3D Dance Generation”. In: *CoRR* abs/2101.08779 (2021). arXiv: 2101.08779. URL: https://arxiv.org/abs/2101.08779.