M2C: Concise Music Representation for 3D Dance Generation

Matthew Marchellus and In Kyu Park
Department of Electrical and Computer Engineering, Inha University
Incheon 22212, Korea
{marchellusmatthew@gmail.com, pik@inha.ac.kr}

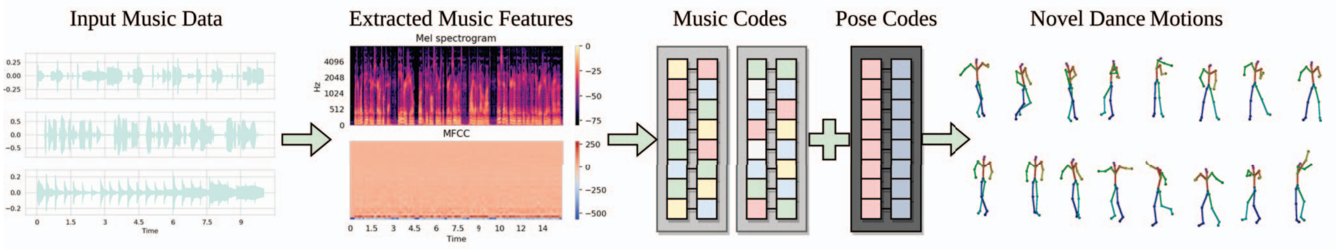


Figure 1: **Music-to-Codes (M2C).** We introduce music codes, a novel music representation for generating music-controlled dance. Moreover, we propose the M2C to formulate music codes and the SM-GPT network to predict dance motions using music codes. Our evaluation result shows that designing the network around music codes improves dance motion quality.

Abstract

Generating 3D dance motions that are synchronized with music is a difficult task, as it involves modelling the complex interplay between musical rhythms and human body movements. Most existing approaches focus on improving the dance generation network, often overlooking the importance of the music feature processing stage which plays a crucial role in dance motion generation. In this paper, we propose music codes, a better latent representation for music features using discrete variables. We present a comprehensive analysis of the music features and propose a different normalization procedure to address the scale imbalance issue within music features. We also introduce the Music-to-Codes (M2C) network, a VQ-VAE inspired network as a music code extractor to replace existing music feature processors. To evaluate the effectiveness of our approach, we combine M2C with Stochastic Motion GPT (SM-GPT), our modification of a recent SoTA dance generation method. Our extensive evaluation and ablation study demonstrates that our dance generation pipeline (using M2C and SM-GPT) significantly improves the dance generation result both qualitatively and quantitatively across all evaluation metrics. Our work opens up new possibilities for exploring the relationship between music and dance, contributing to more effective music-conditioned 3D dance generation.

1. Introduction

Throughout human history, dancing has always been a part of human life. The oldest evidence of dancing as a part of human culture originates in more than thousands of years [32]. Dancing itself can be defined as a human expression of their creativity by manipulating the human body in time and space [19]. Fast forward to the present, dancing still plays a part in our daily life. Particularly, dancing is now an integral part of pop culture and is even more so due to the rise of video-sharing apps such as TikTok and YouTube which houses countless dancing videos accompanied by catchy pop music.

In today's world, the latest advancement in computer graphics enables people to indulge in computer graphics-aided entertainment, including AR/VR, video games, virtual worlds, social networks, and computer-generated movies. While these services offer a wide range of entertainment choices, incorporating dance into them, especially those based on 3D computer graphics, is challenging. It often involves an intricate human motion capture system and a professional dancer to perform the movements. With the ever-growing media consumption rates [6] and the ever-changing internet trends, maintaining the quality of dance motions in these services becomes a challenge.

Music-conditioned 3D dance generation is a challenging computer vision research that involves generating dance motion from a short dance sequence and music input. Particularly, this task is complicated due to human kinematic constraints and requires artistic creativity. In fact, choreographing a high-quality dance motion to accompany a piece of music is a learned and trained skill usually done by a professional choreographer. Prior works approached this problem as a sequence generation task by using a deep multi-modal model to learn the interdependence between the two inputs data modality (*i.e.*, music data and conditioning dance motion) and the supposedly subsequent dance motion [1, 25, 28, 29, 30, 46]. Each of these methods proposes a distinct dance generation method, employing a delicately designed deep neural network. However, a common trait among prior methods is their failure to explicitly leverage on the hidden information contained in the input music feature through their network design.

In this paper, we propose a concise music representation designed to benefit the deep dance generation model by capturing essential features within the music input (Figure 1). We label our proposed feature representation as *music codes*, which is a pair of discrete latent codes. In addition, we present our findings regarding the necessity and benefit of replacing music features with music codes. We provide supporting evidence showing the massive scale imbalance of commonly used extracted music features *mel-frequency cepstral coefficients (MFCC)*.

We propose a Music-to-Codes network for music code formulation given a music feature sequence (inspired by VQ-VAE [51] and 3D Pose VQ-VAE [46]). We design M2C as a discrete autoencoder and train it before using it to formulate music codes for dance generation. Subsequently, we combine M2C's feature extractor with a modified SoTA deep dance generation network, Li *et al.*'s Motion GPT [46], for generating dance movements. The overall combined structure of the M2C and SM-GPT network is illustrated in Figure 2. The modification includes ensuring compatibility with music codes and incorporating a stochastic sampling module for increasing diversity. Despite its straightforward design, the experiment result demonstrates that M2C successfully formulates insightful discrete music codes that enhances the generated dance movements quality. To summarize, our contributions are fourfold.

- We present a comprehensive analysis of the fundamental aspects of music features, specifically highlighting their application in music-conditioned dance generation.
- We propose the M2C network to formulate our novel music code. Despite its simple design, M2C has demonstrated remarkable proficiency in formulating music codes, as evidenced by experimental results.
- To ensure compatibility with music codes, we have made minor modifications to a prior SoTA dance predictor. Additionally, we incorporated a stochastic sampling module to increase the generated dance movements diversity.
- We perform extensive evaluation to validate our proposed music codes, the M2C network, and the dance generation pipeline.

2. Related Works

Motion Generation and Music-Conditioned Dance Generation. Extensive research has been conducted on synthesizing human motion for many years, with early works employing a motion graph-based approach for synthesizing human motion [23, 5, 26, 24]. These approaches synthesize human motion by combining motion graphs, which are made by splitting full recorded motion sequences into sub-sequences. However, music-conditioned dance generation requires a cross-modal understanding between dance motion and musical attributes to generate not just plausible dance sequences, but also matches the given music piece. Recent works have used a learning based method for dance generations such as CNN [27, 10], RNN [44, 49, 4], GAN [48, 1, 41], and transformers network [46, 30, 28, 29, 14] to learn the correlation between musical attributes and corresponding dance moves.

Li *et al.* [30] proposed a dance generation method using a transformer network with full-attention mask, contrary to other prior work (*i.e.*, causal attention mask [29]), and created a large dance motion dataset AIST++ with 3D and 2D joints annotation alongside with music and dance genre annotations. Inspired by traditional animation technique, Li *et al.* [28] proposed another transformer-based dance generation network designed to predict the key-frame for each beat first and then interpolate in-between key-frames by using TCB spline [22] to represent the dance motions. One of the recent works, Li *et al.* [46] proposed a two-stage dance generation network to learn a robust discrete representation of the human pose first, then predict the discrete representation instead of directly tampering with each human joint. As the most recent work, Tseng *et al.* [50] proposed an editable dance generation from music using a diffusion-based approach while leveraging a prior hand-crafted audio feature extraction method [9] to extract music features.

Among the aforementioned methods, many resorted to Librosa [18] to obtain musical attributes as a component of their network input [46, 30, 29, 14, 49, 48]. Contrary to our proposed music code idea, these methods directly fed the extracted music features from Librosa [18] onto their network to learn the appropriate mapping between said music feature and dance motion. On the other hand, Tseng *et al.*'s proposed method [50] shares a similarity with our work as they also leverage a strong input music feature extraction method [9]. However, they neither conduct further analysis for the effectiveness behind using the strong audio feature extractor or use discrete features.

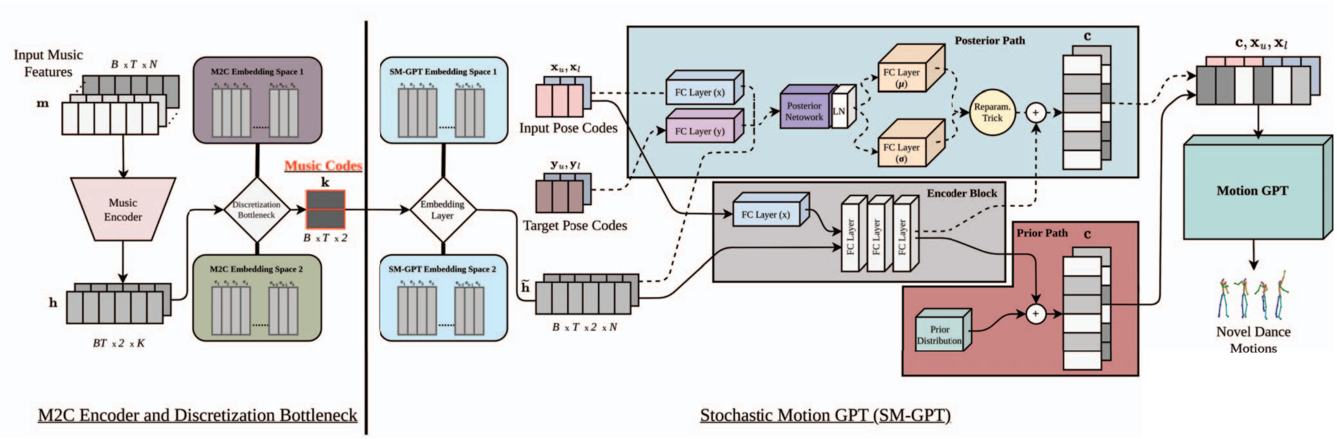


Figure 2: **Complete network pipeline including M2C and SM-GPT.** We illustrate the combined network architecture of our proposed method M2C alongside the dance generator network SM-GPT(derived from Motion GPT [46]). M2C formulates discrete music code sequence $\mathbf{k}$ before feeding it to SM-GPT. SM-GPT maps $\mathbf{k}$ into multidimensional feature $\widetilde{\mathbf{e}}_k$ within their codebooks, and creates the conditional input $\mathbf{c}$ through the residual sampling module. Motion GPT then generates novel dance motion using $\mathbf{c}$ alongside $\mathbf{x}_u$ and $\mathbf{x}_l$ from their 3D Pose VQ-VAE. We train M2C and SM-GPT separately.

Discrete Latent Representation. One variation of latent features is continuous latent variables created using an encoding network (*e.g.*, variational autoencoder [42, 20]). Discrete latent representation is more typically used for language processing tasks given the discrete nature of language (*i.e.*, word tokens) [38, 3]. With that said, some recent computer vision works successfully designed a method which utilizes discrete latent space. Razavi *et al.* [40] designed a multiscale hierarchical VQ-VAE to generate high-quality images, rivalling modern GANs without the notorious GAN training difficulty. Ramesh *et al.* [39] proposed a zero-shot text-to-image method using discrete VAE as the image autoencoder. They also showed that discrete VAE reduces the context size of the transformer network by a factor of 192. Lastly, Li *et al.* [46] utilized the VQ-VAE design within their Motion-VQ network to encode multiple human poses into two pose codes.

Modelling Music Feature with VQ-VAE. There exists some prior work that leveraged VQ-VAE [51, 40] for audio data [9, 7]. Dhariwal *et al.* [9] utilized multiple separate VQ-VAE models (inspired by Hierarchical VQ-VAE [40]) to model different temporal resolutions for the novel music generation task. Bitton *et al.* [7] proposed a generative model based on VQ-VAE [51] which disentangled loudness and learned to quantize a given timbre distribution. Our proposal, the M2C network, shares similarities with these prior methods as it is designed based on VQ-VAE to extract discrete representation by quantizing music features. Yet, unlike them, the motivation behind M2C is to formulate a music code sequence rather than leveraging the quantized encoded features. This is evident as we use a trainable codebook within the SM-GPT to learn a more appropriate feature representation from each music code (refer to Sec. 3 for more details).

3. Proposed Method

We propose the M2C network to learn the discrete mapping of music codes using a set of music features. In practice, the M2C network serves as a substitution module that can replace any music feature encoder module (*e.g.*, FC layer and 1D convolutional layer) in any music-conditioned dance generation method. This is possible as it has no specific requirements and can fit any music feature combination, a versatile option for replacing existing music feature encoders. However, using the M2C network for dance generation requires another network designed specifically for dance generation. To this end, we combine the M2C network with SM-GPT, a modified version of the SoTA dance generation network, Motion GPT [46]. Figure 2 illustrates the complete pipeline of our evaluation network, which includes both M2C and the SM-GPT.

3.1. Understanding Music Features

As a key component for the dance generation, we describe our findings for the music feature analysis. First, we observe that prior dance generation works [46, 30, 28, 29, 25, 50] utilize Librosa [18] to extract music features. Librosa is a public audio-processing toolbox, a common music feature extraction tool for the dance generation task (except for one method [1]). Each dance generation method devises a specific music feature combination as music input.

In Table 1, we show a comprehensive comparison of music feature combinations. We can infer that *MFCC* is a

Method Name	Music Features from Librosa [18]			
	MFCC	MFCC Delta	Chroma	Others
Bailando [46]	Yes	Yes	No	Constant-Q Chromagram, Onset Strength, Tempogram
Li *et al.* [30]	Yes	No	Yes	One-Hot-Peaks, One-Hot-Beats
Danceformer [28]	Yes	No	Yes	-
TSMT [29]	Yes	Yes	No	Beat Interval
Dancing-to-Music [25]	Yes	Yes	No	MFCC Log Energy

Table 1: **Input music features comparison**. Most dance generation methods [25, 29, 28, 30, 46] use Librosa [18] for music feature extraction. Despite having different specifications for their input music feature, every method includes *MFCC)* in their music input. Other features are utilized sparingly according to the network's design.

key input feature for dance generation methods, followed by *MFCC delta* and *Chroma*. *MFCC* extracts the timbre texture feature or spectrum features and is widely used for speech-processing tasks [16, 13, 2, 21, 11]. The extracted timbral characteristics are also particularly useful for music-processing tasks [33, 36, 17, 35]. However, *MFCC* does not construe rhythmic-related features. Therefore, most dance generation methods [46, 30, 28, 29, 25] pair it with other music features.

In Figure 3, we show that the nature of music feature value distribution is unsuitable to be fed to a deep learning network directly. For example, the first coefficient of *MFCC* is the offset value, which has a significantly different mean and standard deviation from the rest of the *MFCC*s. The value scaling imbalance puts more significance into the first five *MFCC*s compared with the other coefficients. This does not align with the deep learning principle as value normalization is not only significant but is often the decisive factor to the method's performance [43, 15, 45].

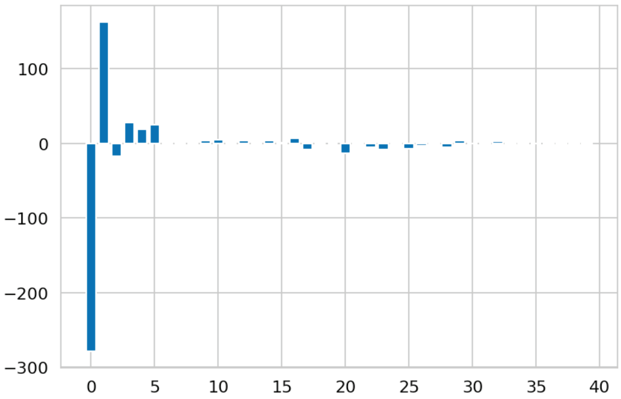


Figure 3: **Average value of each *MFCC* coefficients.** We present the average value of each coefficient in a bar chart. We took the *MFCC* result of three sample music from the Librosa tool [18], namely "choice", "libri1", and "sweetwaltz". Note that due to the massive scale imbalance, some bars are missing for some coefficients (more noticeable after the fifth coefficient)

To alleviate this, we normalize the music features according to their index in the feature vector (we refer to this normalization method as the new norm). Note that this deviates from the standard normalization technique that normalizes features throughout the whole vector dimension. We formulate our music feature normalization method as:

$$\mathbf{m}_{norm} = \left\{ \left\{ \frac{\mathbf{m}_{d,n} - \underset{i\in N}{Mean}(\mathbf{m}_{d,i})}{\underset{i\in N}{Std}(\mathbf{m}_{d,i})} \right\}_{d=0}^{D} \right\}_{n=0}^{N} \tag{1}$$

where $\mathbf{m}$ is the music feature, D is the total dimension of the music feature vector, N is the sum of the music sample and music sequence amount within the train or test dataset, and $Mean(\mathbf{m}_d)$ and $Std(\mathbf{m}_d)$ refers to the mean and standard deviation of each music feature within the specific index d. Our proposed normalization function achieves a uniform value distribution, which resolves the scale issue observed in the standard normalization technique that remains across feature vector samples after normalization.

3.2. Music-to-Codes Network

We design the Music-to-Codes network (M2C) to address the issue mentioned in Sec. 3.1. Inspired by VQ-VAE [51], M2C consists of an encoder block $E(x)$, which encodes the music feature sequence $\mathbf{m} = \langle m_t \rangle_{t=1}^{T}$ into a sequence of encoded feature pairs $\mathbf{h} = \langle h_t^1, h_t^2 \rangle_{t=1}^{T}$, a discretization bottleneck that quantizes $\mathbf{h} \mapsto \mathbf{e}_k$ to the nearest vector $\mathbf{e}_k = \langle e_{k_t}^1, e_{k_t}^2 \rangle_{t=1}^{T}$ within the codebook pair $C = \{\{e_k\}_{k=1}^{K}\}_{n=1}^{N=2}$, and a decoder block $D(x)$ to decode $\mathbf{e}$ into the reconstructed music feature sequence $\hat{\mathbf{m}}$ (illustrated in Figure 4). M2C encodes only the spatial domain of $\mathbf{m}$ while maintaining the temporal domain, resulting in a pair of discrete music code sequences $\mathbf{k} = \langle k_t^1, k_t^2 \in [K] \rangle_{t=1}^{T}$ where K denotes the vocabulary key size. Each codebook $C_{n=1}^{N=2}$ interacts with one of $\mathbf{h}^n$, creating a pair of $\mathbf{e}^n$ and $\mathbf{k}^n$. We first train M2C until convergence to utilize M2C encoder $E(x)$ and codebook C for dance generation.

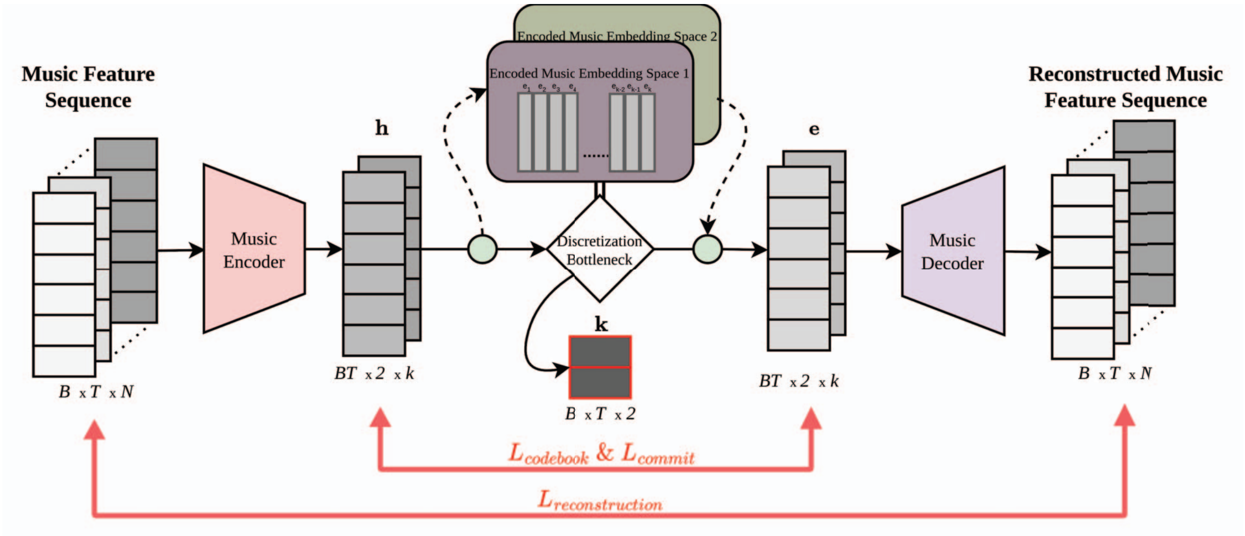

Music Feature Sequence
B x T x N
Music Encoder
h
BT x 2 x k
Encoded Music Embedding Space 2
Encoded Music Embedding Space 1
e1 e2 e3 e4 ...... ek-2 ek-1 ek
Discretization Bottleneck
k
B x T x 2
e
BT x 2 x k
Music Decoder
Reconstructed Music Feature Sequence
B x T x N
L_codebook & L_commit
L_reconstruction

Figure 4: **M2C network architecture**. M2C is a VQ-VAE [51] network designed to formulate music codes. Training M2C is mandatory before using it for training SM-GPT.

We train M2C using the standard reconstruction loss $L_{rec.}$ alongside discretization bottleneck losses [51] (*i.e.*, $L_{codebook}$ and L_{commit}). The reconstruction loss $L_{rec.}(\hat{\mathbf{m}}, \mathbf{m})$ calculates the distance between the reconstructed music feature sequence $\hat{\mathbf{m}}$ and input music feature sequence $\mathbf{m}$. Following VQ-VAE [51], we define the discretization bottleneck losses as

$$L_{codebook} = \frac{1}{T}\sum_{t} ||sg[\mathbf{h}_t] - \mathbf{e}_{k_t}||_2^2 \tag{2}$$

to lessen the distance between the codebook feature $\mathbf{e}_k$ within key k and $\mathbf{h}$ where $sg[.]$ denotes a stop-gradient process and the loss is averaged towards the temporal domain T, and

$$L_{commit} = \frac{1}{T}\sum_{t} ||\mathbf{h}_t - sg[\mathbf{e}_{k_t}]||_2^2 \tag{3}$$

to encourage the encoded feature $\mathbf{h}$ to commit to a certain embedding space, limiting value fluctuation. Combining all of the losses above creates the M2C training objective:

$$L_{M2C} = L_{rec.}(\hat{\mathbf{m}}, \mathbf{m}) + \alpha \times (L_{commit} + L_{codebook}) + \beta \tag{4}$$

where α is the hyperparameter for L_{commit} and $L_{codebook}$, and β is the regularization value.

3.3. Stochastic Motion GPT (SM-GPT)

We leverage Motion GPT [46] as our dance predictor module which will utilize the music codes from M2C. For that reason, we added some modifications towards Motion GPT to ensure compatibility with music codes $\mathbf{k}$ and label the modified version as SM-GPT. We first add a pair of embedding space $\widetilde{C} = \{\{e_k\}_{k=1}^{K}\}_{n=1}^{N=2}$ to map the music codes into multidimensional feature $\mathbf{k} \mapsto \widetilde{\mathbf{e}}_k$. Embedding space pair $\widetilde{C}$ is applied to every $\sum_n^2 \sum_t^T \{\mathbf{k}\}_t^n$, creating a sequence of multidimensional feature pair $\widetilde{\mathbf{e}} = \langle \tilde{e}_t^1, \tilde{e}_t^2 \rangle_{t=1}^T$. This embedding space replaces the dense layer in Motion GPT for input music feature processing. Having a dedicated learnable embedding space within Motion GPT creates a disconnect between $\mathbf{m}$ and $\widetilde{\mathbf{e}}_k$ which creates an opportunity to learn a more appropriate representation for each input music sample based on the extracted music codes.

Furthermore, we incorporate a conditional stochastic sampling module directly after embedding space $\widetilde{\mathbf{C}}$ to improve our dance generation diversity. Similar to the typical CVAE [47], we design a residual stochastic sampling module (labeled as residual sampling) with two main paths, the posterior network for training $q_\theta(\mathbf{z}|\tilde{\mathbf{e}}, \mathbf{x}, \mathbf{y})$ and prior network for testing $p_\theta(\mathbf{z}|\tilde{\mathbf{e}}, \mathbf{x})$. The posterior network $q_\theta(\mathbf{z}|\tilde{\mathbf{e}}, \mathbf{x}, \mathbf{y})$ approximates the posterior distribution over the input dance motion $\mathbf{x}$ and music features $\tilde{\mathbf{e}}$ with the aid of the target dance motion $\mathbf{y}$. Meanwhile, the prior network $p_\theta(\mathbf{z}|\tilde{\mathbf{e}}, \mathbf{x})$ approximates the prior distribution over $\mathbf{x}$ and $\tilde{\mathbf{e}}$ without $\mathbf{y}$, due to the lack of target information $\mathbf{y}$ during test time. Both the posterior network and prior network utilize the same encoder $E_{samp}(x)$ for feature extraction, which is composed of stacked multi-layer perceptrons (MLPs). We utilize the reparameterization trick [20] and sample a noise as input to enable backpropagation while preserving the stochastic sampling aspect. Afterward, a residual connection adds the latent sampling result $\mathbf{z}$ to $E_{samp}(x)$'s result, creating the conditional input $\mathbf{c}$ to Motion GPT alongside the pose code sequence (refer to Li *et al.*'s explanation for further details [46]). We show the overall architecture of SM-GPT in Figure 2.

3.4. Dance Generation with M2C

The problem of music-conditioned dance generation is generating novel dance sequence $\hat{\mathbf{y}} = \langle y_t \rangle_{t=1}^{T}$ given a piece of conditioning dance motion $\mathbf{x} = \langle x_t \rangle_{t=1}^{T}$ and conditioning music features $\mathbf{m} = \langle m_t \rangle_{t=1}^{T}$. As stated previously, we leverage the SM-GPT (explained in Sec. 3.3) which is a modified version of a recent SoTA, Motion GPT [46], as our dance generation network. Furthermore, we utilize a pre-trained M2C to extract music codes $\mathbf{k} = \langle k_t^1, k_t^2 \rangle_{t=1}^{T}$ for the novel dance generation process. During this training step, we only train the SM-GPT and utilize the M2C encoder and discretization bottleneck to extract music codes. Figure 2 illustrates the complete network architecture for novel dance generation using M2C.

To generate novel dance motions, we first extract music codes $\mathbf{k}$ using M2C from the normalized music input $\mathbf{m}_{norm}$ (explained in Sec. 3.1). This is performed using the M2C encoder $E(x)$ to encode the input music feature sequence $\mathbf{m}_t$ into $\mathbf{h}_t^n$, followed by the discretization bottleneck to quantize the encoded features $\mathbf{h}_t^n$ into the quantized features $\mathbf{e}_t^n$ within the respective embedding space $C_{n=1}^{N=2}$ and obtain music codes $\mathbf{k}_t^n$. As stated in Sec. 3.2, the M2C encodes only the spatial domain while leaving the temporal domain intact, maintaining the time step t within the input music feature vector unaffected throughout the music codes extraction process.

SM-GPT transforms the extracted music codes $\mathbf{k}_t^n$ into multidimensional features $\widetilde{\mathbf{e}}_t^n$ using the codebook pair $\widetilde{C}$. This codebook pair $\widetilde{C}$ is a trainable embedding space within SM-GPT to interpret the music codes $\mathbf{k}_t^n$ (not to be confused with M2C's codebook pair C). Having a distinct and dedicated embedding space for each music code introduces a non-linearity and enables the network to learn a more appropriate representation. Afterward, we obtain the conditional input $\mathbf{c}_t$ from the residual sampling module (using the posterior network while training or the prior network while testing). Motion GPT takes the conditional input $\mathbf{c}_t$ and the pose code sequences from the 3D Pose VQ-VAE generate novel dance movements (refer to Li *et al.*'s publication [46] for more explanation).

We adopt Motion GPT's training objective [46] to train the novel dance generation network while adding the variational learning objective Stochastic Gradient Variational Bayes (SGVB) [20] to train the residual sampling module. Assume that $\mathbf{z}$ follows a multivariate Gaussian distribution with a diagonal covariance matrix, we can write the training objective as the evidence lower bound (ELBO)

$$L_{ELBO} = L_{CE} - KL(q_\theta(\mathbf{z}|\tilde{\mathbf{e}}, \mathbf{x}, \mathbf{y}) || p_\theta(\mathbf{z}|\tilde{\mathbf{e}}, \mathbf{x})) \quad (5)$$

where L_{CE} is our reconstruction loss followed by the Kullback-Leibler (KL) divergence between the posterior and the prior distribution. Following Motion GPT, we define L_{CE} as

$$L_{CE} = \frac{1}{T} \sum_{t=1}^{T} \sum_{h=u,l} CrossEntropy(\mathbf{a}_t^h, p_{t+1}^h) \quad (6)$$

where $\mathbf{a}_t$ denotes the future pose code probability given a past pose code $\mathbf{x}_t$ and music code $\mathbf{k}_t^n$, $\mathbf{p}_{t+1}$ denotes the GT future pose code, u, and l refers to the upper or lower body pose code, respectively. We incorporate the *KL cost annealing* [8] which allows the network to train for accuracy by gradually increasing the KL term weight (from 0 to 1).

4. Experimental Results

4.1. Dataset

We train and test our method by using the AIST++ dataset proposed in [30]. AIST++ is a large dataset that contains paired dance and music data, comprising 10 different dance genres. In total, the AIST++ dataset contains 992 3D dance motion sequences sampled at 60 FPS. The duration for each dance motion sequence varies from 7.4s to 48s. Out of the 992 motion sequences, 952 were used for training, and the remaining 40 are for testing. For our method training process, we simply utilize the paired music and dance data and disregard the other annotations (*e.g.*, RGB video, and different angled dance data).

4.2. Implementation Details

In this work, we leverage the human pose structure (human joint graph) parametric 3D body format SMPL [31] as the body model. This is mainly due to our decision to combine M2C with Motion GPT [46] (it utilizes SMPL as its 3D body model). Thus, the input and output pose dimensions are 24-dim of human joints along with a 3-dim global translation vector, resulting in a 219-dim vector. M2C handles 30 samples of music data, each comprising exactly one sample of music data using Librosa [18]. We utilize five categories of music features *MFCC*, *MFCC delta*, *constant-Q chromagram*, *onset strength*, and *tempogram*. Combining all the music features leaves us with 438-dim of music features $\mathbf{m}$. M2C encodes the 438-dim music features into a pair of 55-dim music features $\mathbf{h}$. Thus, we set the number of K within M2C and SM-GPT (C and $\widetilde{C}$ respectively) to 55-dim to match the encoded music feature's dimension. We leverage multiple layers of 1D convolutional and 1D Conv-ResNet within the M2C encoder and decoder. The M2C decoder simply reverses the order of layers within the M2C encoder. We first train M2C to learn a robust understanding of the music feature before using M2C as a pre-trained music code extractor while training SM-GPT. Training M2C using NVIDIA RTX A6000 takes 2.5h while training the M2C network alongside SM-GPT takes roughly 16.7h. We observe that training SM-GPT with M2C increases GPU memory usage by around 2.5% (from 23GB to 23.6GB).

4.3. Evaluation Metrics

We evaluate our method M2C and SM-GPT quantitatively by calculating the accuracy and diversity metric between our generated dance motion and ground truth dance motions. For an accurate evaluation, we adhere to the evaluation protocol from prior works [30, 46, 29, 14, 52] by evaluating the distribution of the generated dance motion via the *Fréchet Inception Distance* (FID). We utilize only one motion feature extractor [12] following a recent work [46] (contrary to using two feature extractor [30]). We perform the quantitative evaluation on two front, the geometric aspect (FID_g and DIV_k) [34] for evaluating the local motion involving certain body parts, and the kinetic aspect (FID_k and DIV_k) [37] for measuring the kinetic attributes (*e.g.*, global velocity, acceleration). We obtain diversity score from the average distance within feature space between generated dance motions [30].

4.4. Method Evaluation and Comparison

We evaluate our method through a quantitative comparison with SoTA prior works [29, 52, 14, 30, 46, 50]. The evaluation is performed using the predetermined AIST++ test set containing 40 samples of dance and music pieces. Moreover, we report the quantitative score of the ground truth data AIST++ for comparison [46]. We compile the quantitative scores from our method, prior works, and ground truth data in Table 2. We presented two variations of our method (*i.e.*, w/o new norm achieves better score, yet qualitatively worse compared to w/ new norm).

We observe the quantitative scores in Table 2 and determine that our method achieves higher accuracy- and diversity-based scores than most prior state-of-the-art (SoTA) works. Moreover, our method gains a significant score improvement from Bailando, which indicates that our method improves the dance generation capability of the target dance generator network. For accuracy metric (*i.e.*, FID_k and FID_g), our method outperforms the prior SoTA and our baseline model [46] with an average of 23% score improvement (at least 35% better FID_k and 10% better FID_g). This is evident within the qualitative comparison as our dance motions do not jitter as much as the baseline method. Furthermore, our method is much closer to the ground truth score (*i.e.*, at least 6% worse FID_k and 19% worse FID_g), which suggests that our method generates dance motion at a similar quality level to the dataset.

Likewise, our method achieves higher diversity scores than most prior SoTAs. Without the new norm, we outperform the baseline and prior SoTA diversity result (*i.e.*, 6% and 5% better DIV_k and DIV_g, respectively) and almost outperforms the dataset itself (*i.e.*, 1% better DIV_k yet loses with 10% worse DIV_g). This is on top of attaining higher accuracy-based evaluation scores, which means that our method successfully improves upon the baseline and produces diverse dance sequences while remaining plausible and accurate.

Method Name	Accuracy ↓		Diversity ↑	
	FID_k	FID_g	DIV_k	DIV_g
GT (AIST++)	17.10	10.60	8.19	7.45
Li *et al.* [29]	86.43	43.46	6.85*	3.32
DanceNet [52]	69.18	25.49	2.86	2.85
Huang *et al.* [14]	73.42	25.92	3.52	4.87
FACT [30]	35.35	22.11	5.94	6.18
Bailando [46]	28.16	9.62	7.83	6.34
EDGE [50]	-	23.08	**9.48**	5.72
M2C +SM-GPT +new norm	18.09	8.62	6.80	5.82
M2C +SM-GPT	**14.68**	**6.04**	8.30	**6.64**

Table 2: **Quantitative evaluation between music conditioned dance generation methods**. Comparison is done using the AIST++ dataset test set. The best and second best results are presented in **bold** and underline, respectively. We obtain the quantitative result from their respective publications, or a re-evaluation result [30, 46]. *As stated by prior works, their generated dance motions are highly jittery, resulting in high-velocity variation.

However, we observe that the qualitative result from our method w/ new norm is visibly superior compared to w/o new norm. Despite ranking lower in the quantitative comparison, our method w/ new norm produces more lifelike dance motion and does not always repeat the same motion throughout the entire generated dance sequence. For more explanation, refer to the supplementary material.

4.5. Ablation Studies

In this section, we perform multiple experiments to determine the effectiveness of our proposed method and their respective key designs.

Effectiveness of our contribution. We report the results of our ablation experiment on M2C and SM-GPT in Table 3. We show that adding res. sampling increases the diversity to some degree at the cost of some accuracy. Moreover, adding M2C is a huge quantitative boost to the quantitative score across the board (except for DIV_g). However, we found that utilizing the new norm often reduces the quantitative result of our method (except for the base Motion GPT network). This is evident in our quantitatively best method as it utilizes both M2C and res. sampling without a new norm. Despite that, combining all together (*i.e.*, M2C, res. sampling, new norm) achieves a balance between good quantitative score and qualitative score. Refer to the supplementary material for more detailed network ablation.

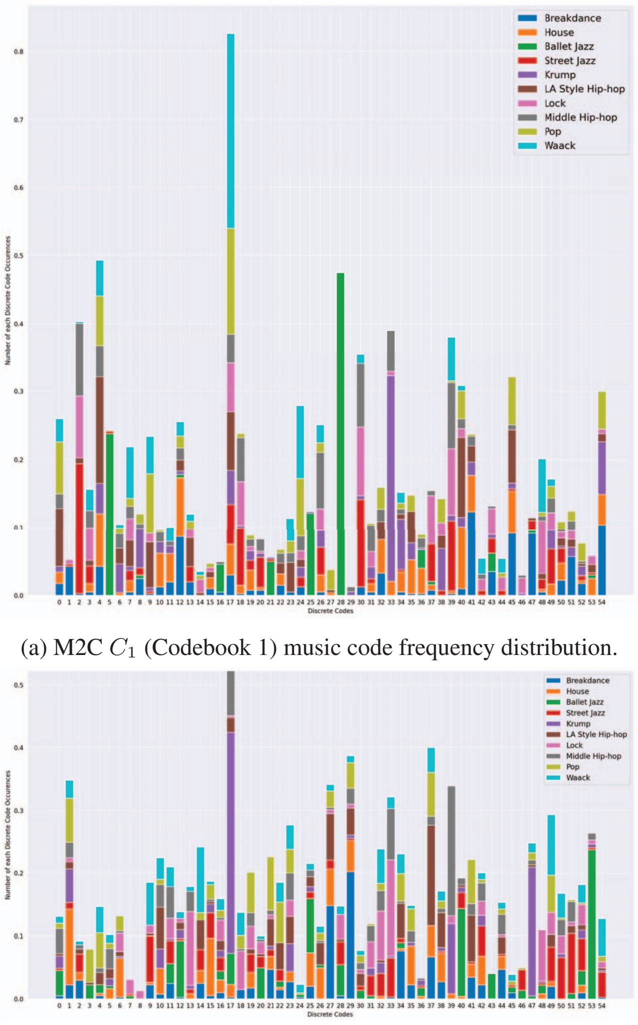
M2C Understanding of Music Features. Figure 5 shows the frequency of extracted music code from the music sequences within AIST++ [30] test set. For each dance genre, we calculate the percentage of each music code's appear-

	Method Name	Accuracy ↓		Diversity ↑	
		FID_k	FID_g	DIV_k	DIV_g
	GT (AIST++)	17.10	10.60	8.19	7.45
	Bailando [46]	28.16	9.62	7.83	6.34
SM-GPT	Motion GPT	66.02	35.62	2.89	3.55
	+new norm	59.99	21.31	3.74	3.74
	+res. sampling	73.81	61.76	3.61	**9.89**
	+res. sampling & +new norm	81.23	65.87	2.79	<u>7.74</u>
+M2C	+new norm & -res. samp.	38.14	11.58	<u>7.11</u>	6.46
	-new norm & +res. samp.	**14.68**	**6.04**	**8.30**	6.64
+All	Ours Best	<u>18.09</u>	<u>8.62</u>	6.80	5.82

Table 3: **Ablation study of the proposed M2C network.** Best and second best results are presented in **bold** and <u>underline</u>, respectively.

ance towards the length of the music code sequence. We then plot a stacked bar chart where we stack said percentage from every music sequence belonging to every dance genre within the AIST++ test set. Thus, the *y*-axes shows the music code distribution within the AIST++ test set.

We observe that within said figure, there is a visible tendency for music belonging to each genre to utilize a specific music code. The code 28 within C_1 is only used for Ballet Jazz music for example, while code 17 in both C_1 and C_2 is the most utilized code, shared by many dance genres. Thus, based on Figure 5, we can conclude that the formulated music codes contain some genre-related information to benefit the dance generation process. Note that the supplementary material contains more detailed comparison regarding other M2C network design with more detailed figures.



(a) M2C C_1 (Codebook 1) music code frequency distribution.

(b) M2C's C_2 (Codebook 2) music code frequency distribution.

Figure 5: **M2C's code distribution within each codebook.** We compile the frequency of music codes for each music sequence within the AIST++ test set.

5. Conclusion

In this paper, we proposed music codes, a novel discrete music representation to avoid the music feature value distribution disparity. Furthermore, music codes create a feature disconnect, allowing the dance generation network to learn more informative features based on each music code. As a proof, we showed that the *MFCC* value disparity indicates a highly different value scale between the initial five and the other coefficients, which is not an issue for music codes due to their discrete nature. In addition, we propose M2C network and SM-GPT to formulate music codes and predict novel dance motions based on music codes, respectively. Experimental results (including ablation study) using M2C and SM-GPT demonstrated that our proposed network achieves its goal to formulate informative music codes and generate high-quality dance motions. Further research can explore the possibilities of integrating music codes into the dance generator network, creating a single end-to-end music-conditioned dance generator method.

Acknowledgement

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.RS-2022-00155915, Artificial Intelligence Convergence Innovation Human Resources Development (Inha University) and No.2022-0-00981, Foreground and Background Matching 3D Object Streaming Technology Development and No.2021-0-02068, Artificial Intelligence Innovation Hub).

References

[1] Hyemin Ahn, Jaehun Kim, Kihyun Kim, and Songhwai Oh. Generative autoregressive networks for 3D dancing move synthesis from music. *IEEE Robotics and Automation Letters*, 5:3501–3508, 2020. 2, 3

[2] Ooi Chia Ai, M Hariharan, Sazali Yaacob, and Lim Sin Chee. Classification of speech dysfluencies with MFCC and LPCC features. *Expert Systems with Applications*, 39(2):2157–2165, 2012. 4

[3] Alan Akbik, Duncan Blythe, and Roland Vollgraf. Contextual string embeddings for sequence labeling. In *Proc. International Conference on Computational Linguistics*, pages 1638–1649, 2018. 3

[4] Omid Alemi, Jules Françoise, and Philippe Pasquier. Groovenet: Real-time music-driven dance movement generation using artificial neural networks. 08 2017. 2

[5] Okan Arikan and David A. Forsyth. Interactive motion generation from examples. *ACM Trans. on Graphics*, 21(3):483–490, 2002. 2

[6] Anne Austin, Jonathan Barnard, Nicola Hutcheon, and David Parry. Media consumption forecasts 2015. 2015. 1

[7] Adrien Bitton, Philippe Esling, and Tatsuya Harada. Vector-quantized timbre representation. *CoRR*, abs/2007.06349, 2020. 3

[8] Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, Andrew M. Dai, Rafal Józefowicz, and Samy Bengio. Generating sentences from a continuous space. In *Proc. Computational Natural Language Learning*, pages 10–21, 2016. 6

[9] Prafulla Dhariwal, Heewoo Jun, Christine Payne, Jong Wook Kim, Alec Radford, and Ilya Sutskever. Jukebox: A generative model for music. *CoRR*, abs/2005.00341, 2020. 2, 3

[10] João Pedro Moreira Ferreira, Thiago M. Coutinho, Thiago L. Gomes, José F. Neto, Rafael Azevedo, Renato Martins, and Erickson R. Nascimento. Learning to dance: A graph convolutional adversarial network to generate realistic dance motions from audio. *Computer Graphics*, 94:11–21, 2021. 2

[11] Todor Ganchev, Nikos Fakotakis, and George Kokkinakis. Comparative evaluation of various MFCC implementations on the speaker verification task. In *Proc. International Conference on Speech and Computer*, number 2005, pages 191–194, 2005. 4

[12] Deepak Gopinath and Jungdam Won. fairmotion - tools to load, process and visualize motion capture data. Github, 2020. 7

[13] Wei Han, Cheong-Fat Chan, Chiu-Sing Choy, and Kong-Pang Pun. An efficient MFCC extraction method in speech recognition. In *Proc. IEEE International Symposium on Circuits and Systems*, 2006. 4

[14] Ruozi Huang, Huang Hu, Wei Wu, Kei Sawada, Mi Zhang, and Daxin Jiang. Dance revolution: Long-term dance generation with music via curriculum learning. In *Proc. International Conference on Learning Representations*, 2021. 2, 7

[15] Xun Huang and Serge J. Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proc. IEEE International Conference on Computer Vision*, pages 1510–1519, 2017. 4

[16] Chadawan Ittichaichareon, Siwat Suksri, and Thaweesak Yingthawornsuk. Speech recognition using MFCC. In *Proc. International Conference on Computer Graphics, Simulation and Modeling*, volume 9, 2012. 4

[17] Jesper Højvang Jensen, Mads Græsbøll Christensen, Manohar N Murthi, and Søren Holdt Jensen. Evaluation of MFCC estimation techniques for music similarity. In *Proc. European Signal Processing Conference*, pages 1–5, 2006. 4

[18] Yanghua Jin, Jiakai Zhang, Minjun Li, Yingtao Tian, Huachun Zhu, and Zhihao Fang. Towards the automatic anime characters creation with generative adversarial networks. *ArXiv*, abs/1708.05509, 2017. 2, 3, 4, 6

[19] Adrienne Kaeppler. Dance in anthropological perspective. *Annual Review of Anthropology*, 7:31–49, 11 2003. 1

[20] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *Proc. International Conference on Learning Representations*, 2014. 3, 5, 6

[21] KV Krishna Kishore and P Krishna Satish. Emotion recognition in speech using MFCC and wavelet features. In *Proc. International Advance Computing Conference*, pages 842–847, 2013. 4

[22] Dirk Kochanek and Richard H. Bartels. Interpolating splines with local tension, continuity, and bias control. In *Proc. SIGGRAPH*, pages 33–41, 1984. 2

[23] Lucas Kovar, Michael Gleicher, and Frédéric H. Pighin. Motion graphs. *ACM Trans. on Graphics*, 21(3):473–482, 2002. 2

[24] Alexis Lamouret and Michiel van de Panne. Motion synthesis by example. In *Computer Animation and Simulation*, pages 199–212, 1996. 2

[25] Hsin-Ying Lee, Xiaodong Yang, Ming-Yu Liu, Ting-Chun Wang, Yu-Ding Lu, Ming-Hsuan Yang, and Jan Kautz. Dancing to music. In *Proc. Advances in Neural Information Processing Systems*, pages 3581–3591, 2019. 2, 3, 4

[26] Jehee Lee, Jinxiang Chai, Paul S. A. Reitsma, Jessica K. Hodgins, and Nancy S. Pollard. Interactive control of avatars animated with human motion data. *ACM Trans. on Graphics*, 21(3):491–500, 2002. 2

[27] Juheon Lee, Seohyun Kim, and Kyogu Lee. Listen to dance: Music-driven choreography generation using autoregressive encoder-decoder network. *CoRR*, abs/1811.00818, 2018. 2

[28] Buyu Li, Yongchi Zhao, Shi Zhelun, and Lu Sheng. Danceformer: Music conditioned 3D dance generation with parametric motion transformer. In *Proc. AAAI Conference on Artificial Intelligence*, pages 1272–1279, 2022. 2, 3, 4

[29] Jiaman Li, Yihang Yin, Hang Chu, Yi Zhou, Tingwu Wang, Sanja Fidler, and Hao Li. Learning to generate diverse dance motions with transformer. *ArXiv*, abs/2008.08171, 2020. 2, 3, 4, 7

[30] Ruilong Li, Shan Yang, David A. Ross, and Angjoo Kanazawa. AI choreographer: Music conditioned 3D dance generation with AIST++. In *Proc. IEEE/CVF International Conference on Computer Vision*, pages 13381–13392, 2021. 2, 3, 4, 6, 7

[31] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: A skinned multi-person linear model. *ACM Trans. on Graphics*, 34(6):248:1–248:16, 10 2015. 6

[32] Yashodhar Mathpal. *Prehistoric Rock Paintings of Bhimbetka, Central India*. 1984. 1

[33] M. Müller. *Information Retrieval for Music and Motion*. Springer Berlin Heidelberg, 2007. 4

[34] Meinard Müller, Tido Röder, and Michael Clausen. Efficient content-based retrieval of motion capture data. *ACM Trans. on Graphics*, 24(3):677–685, 2005. 7

[35] Monica S Nagawade and Varsha R Ratnaparkhe. Musical instrument identification using MFCC. In *Proc. International Conference on Recent Trends in Electronics, Information & Communication Technology*, pages 2198–2202, 2017. 4

[36] NJ Nalini and S Palanivel. Music emotion recognition: The combined evidence of MFCC and residual phase. *Egyptian Informatics Journal*, 17(1):1–10, 2016. 4

[37] Kensuke Onuma, Christos Faloutsos, and Jessica K. Hodgins. FMDistance: A fast and effective distance function for motion capture data. In *Proc. Eurographics (Short Papers)*, pages 83–86, 2008. 7

[38] Matthew E. Peters, Mark Neumann, Luke Zettlemoyer, and Wen-tau Yih. Dissecting contextual word embeddings: Architecture and representation. In *The 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1499–1509, 2018. 3

[39] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *Proc. International Conference on Machine Learning*, volume 139, pages 8821–8831, 2021. 3

[40] Ali Razavi, Aäron van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with VQ-VAE-2. In *Proc. Advances in Neural Information Processing Systems*, pages 14837–14847, 2019. 3

[41] Xuanchi Ren, Haoran Li, Zijian Huang, and Qifeng Chen. Music-oriented dance video synthesis with pose perceptual loss. *CoRR*, abs/1912.06606, 2019. 2

[42] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *Proc. International Conference on Machine Learning*, volume 32, pages 1278–1286, 2014. 3

[43] Shibani Santurkar, Dimitris Tsipras, Andrew Ilyas, and Aleksander Madry. How does batch normalization help optimization? In *Proc. Advances in Neural Information Processing Systems*, pages 2488–2498, 2018. 4

[44] Eli Shlizerman, Lucio M. Dery, Hayden Schoen, and Ira Kemelmacher-Shlizerman. Audio to body dynamics. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7574–7583, 2018. 2

[45] Dalwinder Singh and Birmohan Singh. Investigating the impact of data normalization on classification performance. *Applied Soft Computing*, 97(Part B):105524, 2020. 4

[46] Li Siyao, Weijiang Yu, Tianpei Gu, Chunze Lin, Quan Wang, Chen Qian, Chen Change Loy, and Ziwei Liu. Bailando: 3D dance generation by actor-critic GPT with choreographic memory. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11040–11049, 2022. 2, 3, 4, 5, 6, 7, 8

[47] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. In *Proc. Advances in Neural Information Processing Systems*, volume 28, 2015. 5

[48] Guofei Sun, Yongkang Wong, Zhiyong Cheng, Mohan S. Kankanhalli, Weidong Geng, and Xiangdong Li. Deepdance: Music-to-dance motion choreography with adversarial learning. *IEEE Trans. on Multimedia*, 23:497–509, 2021. 2

[49] Taoran Tang, Jia Jia, and Hanyang Mao. Dance with melody: An lstm-autoencoder approach to music-oriented dance synthesis. In *Proc. ACM International Conference Multimedia*, pages 1598–1606, 2018. 2

[50] Jo-Han Tseng, Rodrigo Castellon, and C. Karen Liu. Edge: Editable dance generation from music. *ArXiv*, abs/2211.10658, 2022. 2, 3, 7

[51] Aäron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In *Proc. Advances in Neural Information Processing Systems*, pages 6306–6315, 2017. 2, 3, 4, 5

[52] Wenlin Zhuang, Congyi Wang, Jinxiang Chai, Yangang Wang, Ming Shao, and Siyu Xia. Music2dance: Dancenet for music-driven dance generation. *ACM Trans. on Multimedia Computing, Communications, and Applications*, 18(2):65:1–65:21, 2022. 7