Quantized GAN for Complex Music Generation from Dance Videos

Ye Zhu *
Illinois Institute of Technology

Kyle Olszewski
Snap Inc.

Yu Wu
Princeton University

Panos Achlioptas
Snap Inc.

Menglei Chai
Snap Inc.

Yan Yan
Illinois Institute of Technology

Sergey Tulyakov
Snap Inc.

Abstract

We present Dance2Music-GAN (D2M-GAN), a novel adversarial multi-modal framework that generates complex musical samples conditioned on dance videos. Our proposed framework takes dance video frames and human body motion as input, and learns to generate music samples that plausibly accompany the corresponding input. Unlike most existing conditional music generation works that generate specific types of mono-instrumental sounds using symbolic audio representations (e.g., MIDI), and that heavily rely on pre-defined musical synthesizers, in this work we generate dance music in complex styles (e.g., pop, breakdancing, etc.) by employing a Vector Quantized (VQ) audio representation, and leverage both its generality and the high abstraction capacity of its symbolic and continuous counterparts. By performing an extensive set of experiments on multiple datasets, and following a comprehensive evaluation protocol, we assess the generative quality of our approach against several alternatives. The quantitative results, which measure the music consistency, beats correspondence, and music diversity, clearly demonstrate the effectiveness of our proposed method. Last but not least, we curate a challenging dance-music dataset of in-the-wild TikTok videos, which we use to further demonstrate the efficacy of our approach in real-world applications – and which we hope to serve as a starting point for relevant future research. The code is available at https://github.com/L-YeZhu/D2M-GAN.

1. Introduction

"When the music and dance create with accord, their magic captivates both the heart and the mind." [1] As a natural form of expressive art, dance and music have enriched our daily lives with a harmonious interplay of melodies, rhythms, and movements across the millennia. The growing popularity of social media platforms for sharing dance videos, such as TikTok, has also demonstrated their significance as a source of entertainment in modern society. At the same time, new research is flourishing in the wake of this trend by exploring multi-modal generative tasks linking dance motion and music [1, 37–39].

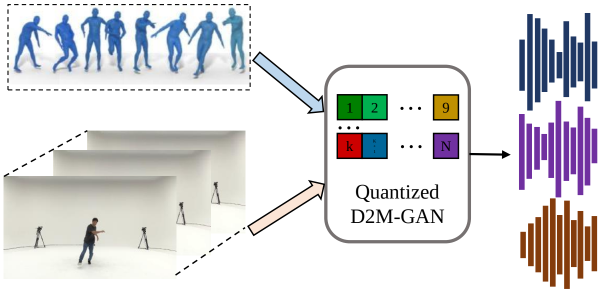


Figure 1. **Task illustration.** We introduce a Vector Quantization framework for music generation from dance videos, which takes human body motion and visual frames as input, and generates suitable corresponding music. Our proposed model can generate complex and rich dance music - in contrast to most existing conditional music generation works, which typically output mono-instrumental sounds.

Although seemingly intuitive, music generation from dance videos has been a challenging task compared to its counterpart in the inverse direction (*i.e.*, dance generation from music) for two primary reasons. First, typical audio music signals are high-dimensional and require sophisticated temporal correlations for overall coherence [4, 28]. For example, CD-quality audio has a typical sampling rate of 44.1 kHz, resulting in over 2.5 million data points ("dimensions") for a one-minute musical piece [9]. In contrast, most dance generation works output the relatively low-dimensional motion data in the form of 2D or 3D skeleton keypoints (*e.g.*, displacements for dozens of joints) conditioned on the music [37, 39, 53, 56], which are then rendered into dance sequences and videos. To tackle the chal-

[*] This work was mainly done while the author was an intern at Snap Inc.

[1] Jean-Georges Noverre.

lenge of the high dimensionality of audio data, research studies on music generation from visual input [16, 25, 57] often rely on low-dimensional intermediate symbolic audio representations (*e.g.*, a 1D piano-roll or 2D MIDI). The symbolic representations provide existing learning frameworks with a more explicit audio-visual correlation mapping and more stable training, as well as widely-established standard music synthesizers for decoding the intermediate representations. However, such symbolic-based works suffer from serious limitations on the flexibility of the generated music. This brings us to the second challenge of dance video conditioned music generation: a separately trained model is usually required for *each* instrument, and the generated music is composed with acoustic sounds from a *single predefined* instrument [12, 16, 46]. Consequently, the resulting music is typically simple, and lacking in harmony and richness consistent with the accompanying real-world dance videos (*e.g.*, see the person dancing in a hip-hop style with piano-based generated samples in our supplementary video). These facts make existing conditional music generation works difficult to generalize into complex musical styles and real-world scenarios.

To fill this gap, we propose a novel adversarial multi-modal framework that learns to generate complex musical samples from dance videos via Vector Quantized (VQ) audio representations. Inspired by the recent success of VQ-VAE [9,45,52] and VQ-GAN [14], we adopt quantized vectors as our intermediate audio representation, and leverage both their increased abstraction ability compared to continuous raw audio signals, as well as their flexibility to better represent complex real-world music in comparison to classic symbolic representations. Specifically, our framework takes the visual frames and dance motion as input (Figure 1), which are encoded and fused to generate the corresponding audio VQ representations. After retrieving the generated VQ representations from a learned codebook, these entries are decoded back to the raw audio domain using a fine-tuned JukeBox decoder [9]. Additionally, we deploy a convolution-based backbone and follow a hierarchical structure with two separate abstraction levels (*i.e.*, different hop-lengths) for the audio signals to demonstrate the scalability of our framework. The higher-level model has a larger hop-length and fewer parameters, resulting in faster inference. In contrast, the lower-level model has a lower abstraction level with smaller hop-length, which enables the generation of music with higher fidelity and better quality.

Last but not least, we also procure a real-world paired dance-music dataset collected from TikTok video compilations. Our dataset contains in total 445 dance videos with 85 songs and an average per-video duration of approximately 12.5 seconds. Unlike existing datasets (e.g., AIST [39,60]), ours is more challenging and better reflects the conditions of real-world scenarios, thus providing a new asset for relevant future research.

Employing such datasets, we conduct extensive experiments to demonstrate the effectiveness and robustness of the proposed framework. Specifically, we design and follow a rich evaluation protocol to consider its generative quality with respect to the correspondence to the input dance motion in in terms of beats, genre and coherence. The general quality of the generated music is also assessed. The attained results (both quantitative and qualitative) demonstrate that our model can generate plausible dance music in terms of various musical features, outperforming several competitive conditional music generation methods.

In summary, our main contributions are:

- We propose *D2M-GAN*, a novel adversarial multi-modal framework that generates complex, free-form music from dance videos via *Vector Quantized (VQ) representations*.
- The proposed model, using a VQ generator and a multi-scale discriminator, is able to effectively capture the temporal correlations and rhythm for the musical sequence to generate complex music.
- To assess our model, we introduce a comprehensive *evaluation protocol* for music conditionally generated from videos, and demonstrate how the proposed *D2M-GAN* generates more complex and plausibly corresponding music compared to existing approaches.
- Last but not least, we create a novel real-world dataset with dance videos captured *in the wild* – and use it to establish a new, more challenging setup for conditioned music generation, which further demonstrates the superiority of our framework.

2. Related Work

2.1. Audio, Vision, and Motion

Combining data from audio, vision, and motion has been a popular research topic in recent years within the field of multi-modal learning. Research focusing on general audio-visual learning typically assumes that the two modalities are intrinsically correlated based on the natural synchronization of the audio and visual signals [2, 3, 34, 47, 48, 67]. Such jointly learned audio-visual representations thus can be applied in multiple downstream tasks, like sound source separation [17–20, 66], audio-visual captioning [51, 62], audio-visual action recognition [21, 31], and audio-visual event localization and parsing [59, 63, 64, 67].

On the other hand, another branch of studies closely related to our work has investigated the correlations between motion and sounds [15,16,37–39,53,56,68]. A large portion of this research aims to generate human motion based on audio signals, either in the form of 2D poses [37, 53, 56] or direct 3D motion [29, 39, 58]. For the inverse direction that seeks to generate audio from motion, Zhao *et al.* [66] introduces an end-to-end model to generate sounds from motion trajectories using a curriculum learning scheme. Gan *et al.* [16] propose a graph-based transformer framework to generate music from performance videos using raw movement as input. Di *et al.* [10] propose to generate video background music conditioned on the motion and special timing/rhythmic features of the input videos. In contrast to these previous works, our work combines three modalities, which takes the vision and motion data as input and generates music accordingly.

2.2. Music Generation

Raw music generation is a challenging task due to the high dimensionality of the audio data and its sophisticated temporal correlations. Therefore, the existing music generation approaches usually adopt an intermediate audio representation for learning generative models to reduce the computational demand and simplify the learning process [9, 12, 25, 35, 44]. Classical audio representations mainly employ the symbolic and continuous approaches. Musegan [12] introduces a multi-track GAN-based model for instrumental music generation via 1D piano-roll symbolic representations. Music Transformer [25] aims to improve the long-term coherence of generated musical pieces using 2D event-based MIDI-like audio representations [46]. Melgan [35] is a generative model for music in form of the audio mel-spectrogram features. Recently, JukeBox [9] introduces a generic music generation model based on the novel Vector Quantized (VQ) representations. Our proposed framework adopts this VQ representation for music generation.

2.3. Vector Quantized Generative Models

VQ-VAEs [45,52] are firstly proposed as a variant of the Variational Auto-Encoder (VAE) [32] with discrete codes and learned priors. Following works have demonstrated the potential of VQ-based framework in multiple generative tasks such as image and audio synthesis [9, 14, 26]. Specifically, the VQ-VAE [45] is initially tested for generating images, videos, and speech. An improved version of VQ-VAE [52] is proposed with a multi-scale hierarchical organization. Esser *et al.* [14] apply the VQ representations in the GAN-based framework for generating high-resolution images. Dhariwal *et al.* [9] introduce the JukeBox as a large-scale generative model for music synthesis based on VQ-VAE. Compared to the classic symbolic and continuous audio representations, the VQ representations leverage the benefits of flexibility (*i.e.*, the ability to represent complex music genres with a unified codebook in contrast to symbolic representations) and high compression levels (*i.e.*, the learned codebooks largely reduce the data dimensionality compared to raw waveform or spectrogram). Our proposed framework combines both the GAN [23] and VAE [32], which uses the GAN-based learning to generate VQ representations from the dance videos, and adopts the VAE-based decoder for synthesizing music.

3. Method

An overview of the architecture of the proposed D2M-GAN is shown in Figure 2. Our approach employs a hierarchical structure with two levels of generative models that are independently trained with a similar pipeline for flexible scalability. In each level, the model consists of four components: the motion module, the visual module, the VQ module consisting of a VQ generator with multi-scale discriminators, and the music synthesizer. Our hierarchical structure provides the flexibility to balance the generated music quality and computational costs given practical application considerations.

3.1. Data Representations

During inference, the input to our proposed *D2M-GAN* comes from two domains: the visual frames of the dance videos and the inferred human body motion of the dancers. The ground-truth audio is also used as the supervision for the discriminators during the training stage. For the human body motion, several different data representations, such as the 3D Skinned Multi-Person Linear model (SMPL) [41] or 2D body keypoints [5, 6] can be employed in our framework. We use SMPL and 2D body keypoints for different datasets in our experiments. To encode the visual frames, we extract I3D features [7] using a model pre-trained on Kinectics [30]. For the musical data, we adopt quantized vectors as the intermediate audio representation. In order to leverage the strong representation ability of codebooks trained on a large-scale musical dataset, we use the codebooks from a pre-trained JukeBox [9] model, which is trained on a dataset of 1.2 million songs.

3.2. Generator

The generator $G = \{G_m, G_v, G_{\text{vq}}\}$ includes the motion module G_m, the visual module G_v, and the principal VQ generator G_{vq} in the VQ module, which takes the fused motion-visual data as input and outputs the desired VQ audio representations.

$$f_{\text{vq}} = G_{\text{vq}}(G_m(x_m), G_v(x_v)) = G(x_m, x_v), \tag{1}$$

where x_m and x_v represent the motion and visual input data, respectively. f_{vq} is the output VQ representations. All these modules are implemented as convolution-based feed-forward networks. For the principal VQ generator, we use leaky rectified activation functions [65] for its hidden layers and a tanh activation for its last layer before output to promote the stability of GAN-based training [50].

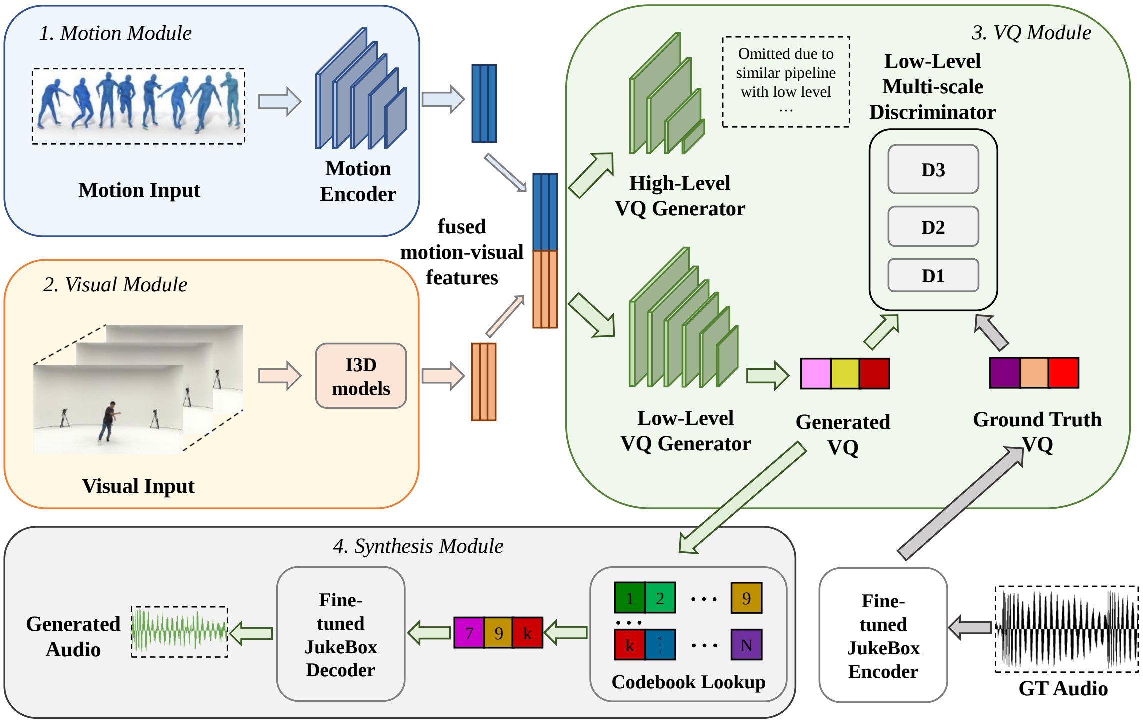


Figure 2. **Overview of the proposed architecture of the *D2M-GAN***. Our model takes the motion and visual data from the dance videos as input and process them with the motion and visual modules, respectively. It then forwards the fused representation containing information from both modalities to ground the generation of audio VQ-based representations with the VQ module. The resulting features are calibrated by a multi-scale GAN-based discriminator and are used to perform a *lookup* in the pre-learned codebook. Last, the retrieved codebook entries are decoded to raw musical samples via by a pre-trained and fine-tuned decoder, responsible for synthesizing music.

It is also worth noting that we find that using batch normalization and the aforementioned activation function designs [42, 50, 55] is crucial for a stable GAN training in our framework. However, the application of the tanh activation will also restrict the output VQ representations within the data range between −1 and +1. We choose to scale activation after the last tanh activation by multiplying by a factor σ. The hyper-parameter σ enlarges the data range of VQ output and makes it possible to perform the lookup of pre-learned large-scale codebooks $\mathrm{LookUp}(f'_{\mathrm{vq}})$ with $f'_{\mathrm{vq}} = \sigma f_{\mathrm{vq}}$. Another significant observation regarding the generator's design is using a wide receptive field. Music has long temporal dependencies and correlations compared to images, therefore, the principal VQ generator with a larger receptive field is beneficial for generating music samples with better quality, which is consistent with the findings from previous works [11, 35]. To this end, we design our generator with relatively large kernel sizes in the convolutional layers, and we also add residual blocks with dilations after the convolutional layers. All previously described submodules within our generator G are jointly optimized.

3.3. Multi-Scale Discriminator

Similar to the generator, the discriminator in the D2M-GAN is also expected to capture the long-term dependencies of musical signals encoded in the generated sequence of VQ features. However, unlike the generator design, which focuses on increasing the receptive fields of the neural networks, we address this problem in the discriminator design by using a multi-scale architecture. The multi-scale discriminator design has been studied in previous works within the field of audio synthesis and generation [33, 35, 61].

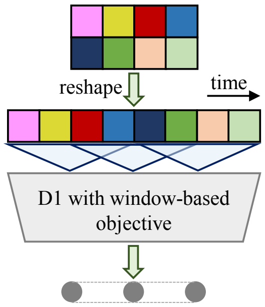


Figure 3. Illustration of the important reshape operation and the window-based discriminator for our *D2M-GAN*.

The discriminator $D = \{D_1, D_2, D_3\}$ in the VQ module of our D2M-GAN is composed of 3 discriminators that operate on the sequence of generated VQ representations and its downsampled features by a factor of 2 and 4, respectively. Specifically, unlike the multi-scale discriminators proposed in previous works that directly take the raw audio as input, we reshape the VQ representations f'_{vq} along the temporal dimension before feeding them into the discriminators, which is also important for *D2M-GAN* to reach a stable adversarial training, as music is a temporal audio sequence. Finally, we use the window-based objectives [35] (Markovian window-based discriminator analog to image patches in [27]). Instead of learning to distinguish the distributions between two entire sequences, window-based objective learns to classify between distributions of small chunks of VQ sequences to further enhance the overall coherence as illustrated in Figure 3.

3.4. Lookup and Synthesis

After generating the VQ representations, we perform a codebook lookup operation similar to other VQ-based generative models [9, 14, 45, 52] to retrieve the closest corresponding entries.

Finally, we fine-tune the decoder from the JukeBox [9] without modifying the codebook entries as the music synthesizer for our learned VQ representations. Specifically, we also adopt the GAN-based technique for fine-tuning the music synthesizer, where the generator is replaced by the decoder of JukeBox and the discriminator follows the similar architecture as described in the previous subsection.

3.5. Training Objectives

GAN Loss. We use the hinge loss version of GAN objective [40, 43] adopted for our music generation task to train the proposed *D2M-GAN*.

$$\begin{aligned} L_{adv.}(D;G) &= \sum_k L_{adv.}(D_k;G) \\ &= \sum_k (\mathbb{E}_{\phi(x_a)}[min(0, 1 - D_k(\phi(x_a)))] \\ &+ \mathbb{E}_{(x_m,x_v)}[min(0, 1 + D_k(G(x_m,x_v)))]), \end{aligned} \tag{2}$$

$$L_{adv.}(G;D) = \mathbb{E}_{x_m,x_v}[\sum_k -D_k(G(x_m,x_v))], \tag{3}$$

where x_a is the original music in a waveform, ϕ represents the fine-tuned encoder from JukeBox [9]. k indicates the number of multi-scale discriminators, which is empirically chosen to be 3 in our case.

Feature Matching Loss. To encourage the construction of subtle details in audio signals, we also include a feature matching loss [36] in the overall training objective. Similar to the audio generation works [33, 35], the feature matching loss is defined as the L_1 distance between the discriminator feature maps of the real and generated VQ features.

$$\begin{aligned} &L_{FM}(G;D) = \\ &\mathbb{E}_{(x_m,x_v)}[\sum_{i=1}^{T} \frac{1}{N_i} \left\| D^i(\phi(x_a)) - D^i(G(x_m,x_v)) \right\|_1]. \end{aligned} \tag{4}$$

Codebook Commitment Loss. The codebook commitment loss [45, 52] is defined as the L_1 distance between the generated VQ features and the corresponding codebook entries of the ground truth VQ features after the codebook lookup process.

$$L_{code}(G) = \mathbb{E}_{(x_m,x_v)}[\| LookUp(\phi(x_a) - G(x_m,x_v)\|_1]. \tag{5}$$

Audio Perceptual Losses. To further improve the perceptual auditory quality, we consider the perception losses of the raw audio signals from both time and frequency domains. Specifically, the perceptual losses are calculated as the L_1 distance between the original audio and the generated audio samples:

$$L_{wav}(G) = \mathbb{E}_{(x_m,x_v)}[\| x_a - G(x_m,x_v)\|_1]. \tag{6}$$

$$L_{Mel}(G) = \mathbb{E}_{(x_m,x_v)}[\| \theta(x_a) - \theta(G(x_m,x_v))\|_1]. \tag{7}$$

where θ is the function to compute the mel-spectrogram features for the audio signal waveforms.

Final Loss. The final training objective for the entire generator module is defined as follows:

$$\begin{aligned} L_G = &L_{adv.}(G;D) + \lambda_{fm} L_{FM}(G;D) \\ &+ \lambda_c L_{code} + \lambda_w L_{wav} + \lambda_m L_{mel}, \end{aligned} \tag{8}$$

where the λ_{fm}, λ_c, λ_a, and λ_{mel} are set to be 3, 15, 40 and 15, respectively during our experiments for both levels.

4. Experiments

4.1. Experimental Setup

Datasets. We validate the effectiveness of our method by conducting experiments on two datasets with paired dance video and music: the AIST++ [39] and our proposed TikTok dance-music dataset. The AIST++ dataset [39] is a subset of AIST dataset [60] with 3D motion annotations. We adopt the official cross-modality data splits for training, validation, and testing, where the videos are divided without overlapping musical pieces between the training and the validation/testing sets. The number of videos in each split is 980, 20, and 20, respectively. The videos from this dataset are filmed in professional studios with clean backgrounds. There are in total 10 different dance genres and corresponding music styles, which include breakdancing, pop, lock, *etc*. The number of total songs is 60, with 6 songs for each type of music. We use this dataset for the main experiments and evaluations.

We also collect and annotate a **TikTok dance-music dataset** which contains 445 dance videos, with an average length of 12.5 seconds. This dataset contains 85 different songs, with the majority of videos having a single dance performer, and a maximum of five performers. The training-testing splits contain 392 and 53 videos, respectively, without overlapping songs. Figure 4 shows example

Figure 4. **Examples of dance videos from *our* TikTok dance-music dataset.** Unlike the AIST dataset [60] where dancing is performed by professional dancers in a studio environment, our dataset consists of real-world videos collected "in the wild".

frames of the dance videos and makes apparent the key differences compared to the professional studio filmed dance video from AIST [60]. Our videos have wildly different backgrounds, and often contain incomplete human body skeleton data, which significantly increases the challenge of learning from this dataset. For the TikTok music dataset, we use 2D human skeleton data as the underlying motion representation.

Implementation Details. For the presented experiments, we adopt a sampling rate of 22.5 kHz for all audio signals. We use the video and audio segments in the length of 2 seconds for training and standard testing in the main experiments. The generation of longer sequences is also investigated in Section 4.3. The hop lengths for the high and low level are 128 and 32, respectively. During the GAN training, we adopt the Adam optimizer with a learning rate of 1e-4 with β_1 = 0.5 and β_2 = 0.9 for the generators and discriminators. We define the scaling factor σ = 100 for the VQ generators. The number of discriminators k is 3 for the multi-scale structure. The batch size is set to be 16 for all experiments. During the fine-tuning of the JukeBox synthesizer, we use the Adam optimizer with a learning rate of 1e-5 with β_1 = 0.5 and β_2 = 0.9 for the synthesizer and multi-scale discriminators. We perform a denoising process [54] on the generated raw music data for better audio quality.

Comparisons. We compare our proposed method with several baselines. *Foley Music [16]*: Foley Music model generates MIDI musical representations based on keypoints motion data and then converts the MIDI back to a raw waveform using a pre-defined MIDI synthesizer. Specifically, the MIDI audio representation is unique for each musical instrument, and therefore the Foley music model can only generate musical samples with mono-instrumental sound. *Dance2Music [1]*: Similar to [16], the generated music with this method is also monotonic in terms of the musical instrument. *Controllable Music Transformer (CMT)* [10]: CMT is a Transformer-based model proposed for video background music generation using MIDI representation. In addition to the above cross-modality models that are closely related to our work, we also consider *Ground Truth:* GT samples are the original music from dance videos. *JukeBox [9]*: music samples generated or reconstructed via the JukeBox model.

4.2. Music Evaluations

We design a comprehensive evaluation protocol that incorporates objective (*i.e.*, metrics that can be automatically calculated) and subjective (*i.e.*, scores given by human testers) metrics to evaluate the generated music from various perspectives. Specifically, the evaluations are divided into two categories: the first category, which is also the focus of our work, measures correlations between the generated music and the input dance videos, for which we compare our proposed model with other cross-modality music generation works [1, 10, 16] and a random baseline from JukeBox [9]. The second category focuses on the quality of the music in general, for which we use the reconstructed samples using JukeBox [9] given the original audio as input and GT samples for comparisons.

Rhythm. Musical rhythm accounts for an important characteristic of the generated music samples, especially given the dance video as input. To evaluate the correspondence between the dance beats and generated musical rhythm, we adopt two objective scores as evaluation metrics, which are the Beats Coverage Scores and the Beats Hit Scores similar to [8, 37]. Previous works [8, 37] have demonstrated the kinematic dance and musical beats (*i.e.*, rhythm) are generally aligned, we can therefore reasonably evaluate the musical rhythm by comparing the beats from the generated music and those from the GT music samples as shown in Fig. 5. We detect the musical beats by the second-level onset strength [13], which can be considered as the start of an acoustic event. We define the number of detected beats from the generated music samples as B_g, the total beats from the original music as B_t, and the number of aligned beats from the generative samples as B_a. The Beats Coverage Scores B_g/B_t measure the ratio of overall generated beats to the total musical beats. The Beats Hit Scores B_a/B_t measure the ratio of aligned beats to the total musical beats. The quantitative results are presented in Table 1. We observe that both levels of our proposed *D2M-GAN* achieve better scores compared to competing methods.

Genre and Diversity. Dance and music are both diverse in terms of genres. The generated music samples are expected to be diverse and harmonious with the given dance style (*e.g.*, breakdancing with strong beats paired with music in fast rhythm). Therefore, we calculate the genre ac-

Category	Features	Type	Metric	Methods	Scores
Dance-Music	Rhythm	Obj.	Beats Coverage & Beats Hit	Dance2Music [1]	83.5 & 82.4
				Foley Music [16]	74.1 & 69.4
				CMT [10]	85.5 & 83.5
				Ours High-level	88.2 & 84.7
				Ours Low-level	**92.3 & 91.7**
Dance-Music	Genre&Diversity	Obj.	Genre Accuracy (Retrieval-based)	Dance2Music [1]	7.0
				Foley Music [16]	8.1
				CMT [10]	11.6
				Ours High-level	24.4
				Ours Low-level	**26.7**
Dance-Music	Coherence	Subj.	Mean Opinion Scores	Random JukeBox [9]	2.0
				Dance2Music [1]	2.8
				Foley Music [16]	2.8
				CMT [10]	3.0
				Ours High-level	3.5
				Ours Low-level	3.3
				GT	**4.6**
Music	Overall quality	Subj.	Mean Opinion Scores	JukeBox [9]	3.5
				Ours High-level	3.5
				Ours Low-level	3.7
				GT	**4.8**

Table 1. Evaluation protocol and the corresponding results for the experiments on the AIST++ dataset [39]. *Obj.* stands for *Objective*, which means the scores are automatically calculated. *Subj.* stands for *Subjective*, which means the scores are given by human evaluators.

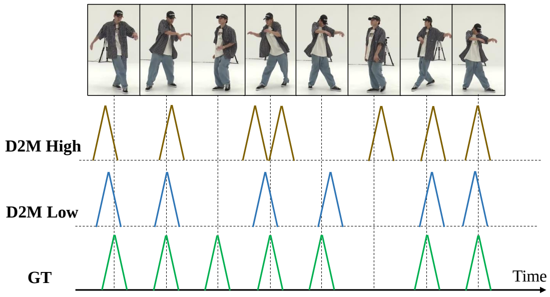


Figure 5. **Qualitative example of rhythm evaluations and beat correspondence.** The lower-abstraction level model (D2M-Low) appears to align better than its high-counterpart (D2M-High) with the ground-truth (GT), which is consistent with the quantitative scores from the Table 1.

curacy for evaluating whether the generated music samples have a consistent genre with the dance style. The calculation of this objective metric requires the annotations of dance and music genres, therefore, we use the retrieved musical samples from the AIST++ [39] for this evaluation setting. Specifically, we *retrieve* the musical samples with the highest similarity scores from the segment-level database formed by original audio samples with the same sequence length. The similarities scores are defined as the euclidean distance between the audio features extracted via a VGG-like network [24] pre-trained on AudioSet [22]. In case that the retrieved musical sample has the same genre as the given dance style, we consider the segment to be genre accurate. The genre accuracy is then calculated by S_c/S_t, where S_c counts the number of genre accurate segments and S_t is the total number of segments from the testing split.

We observe in Table 1 that the genre accuracy scores of our *D2M-GAN* are considerably higher compared to the competing methods. This is due to the reason that the competing methods rely on MIDI events as audio representations, which require a specific synthesizer for each instrument, and thus can only generate music samples with mono-instrumental sound. In contrast, our generated VQ audio representations can represent complex dance music similar to the input music types, which helps to increase the diversity of the generated music samples. It also makes the generated samples to be more harmonious with the dance videos compared to acoustic instrumental sounds from [1,10,16], as shown in the next evaluation protocol for the coherence test.

Coherence. Since we generate music samples conditioned on the dance videos, the dance video input and the output are expected to be harmonious and coherent when combined together. Specifically, a given dance sequence could be accompanied by multiple appropriate songs. However, the evaluation of the dance-music coherence is very subjective, therefore we conduct the Mean Opinion Scores (MOS) human test for assessing the coherence feature. During the evaluation process, the human testers are asked to give a

Models	Beats Coverage	Beats Hit
High w/o M	85.5	72.4
High w/o V	86.3	81.7
High (full)	**88.4**	**82.3**
Low w/o M	83.8	74.6
Low w/o V	85.2	81.7
Low (full)	**87.1**	**83.9**

Table 2. Evaluations for the experiments on the TikTok dataset.

score between 1 and 5 to evaluate the coherence between the dance moves and the music given a video with audio sounds. The higher scores indicate the fact the tester feels the given dance and music are more coherent. We prepare the videos with original visual frames and fused generated music samples for testing. In addition to the previously cross-modality generation methods [1, 10, 16], we also include the GT samples and the randomly generated music from JukeBox [9] for comparison. Our *D2M-GAN* achieves better scores compared to other baselines, which validates the fact that our proposed framework is able to catch the correlations with the given dance video and generates rather complex music that well matches the input.

Overall Quality. Although our main research focus is to learn the dance-music correlations in this work, we also look at the general sound quality of the generated samples. We conduct the subjective MOS tests similar to the coherence evaluation, where the human testers are asked to give a score between 1 to 5 for the general quality of the music samples. During this test, only audio signals are played to the testers. The JukeBox samples are obtained by directly feeding the GT samples as input. The MOS tests show that our *D2M-GAN* is able to generate music sample with plausible sound quality comparable to the JukeBox. JukeBox has multiple variants with different hop lengths, we compare with samples obtained from the model with same audio hop length for fairness (*i.e.*, the hop lengths for our high and low levels are 128 and 32, respectively.). It is worth noting that synthesizing high quality audio itself has been a vary challenging and computational demanding research topic, for example, it takes *3 hrs* to sample a 20-seconds high-quality music sample with a hop length of 8 [9].

Results on the TikTok Dance-Music Dataset. Compared to the AIST++ [39], our TikTok dance-music dataset is a more challenging dataset with "in the wild" video settings that contains various occlusions and noisy backgrounds. Table 2 shows the quantitative evaluation results for the experiments on the TikTok dataset, which demonstrates the overall robustness of the proposed *D2M-GAN*.

4.3. Ablation Studies

Sequence Length. In the main experiments, we use the 2-second length sequence for experiments with reference to other similar cross-modality generation tasks [39]. However, our model can also be effectively trained and tested with a longer sequence length as shown in Table 3 via a relatively larger network with more parameters.

Length	Beats Coverage	Beats Hit	Genre Acc.
High - 2s	**88.2**	84.7	24.4
High - 3s	**88.2**	**85.3**	**25.6**
High - 4s	87.1	83.0	23.3
Low - 2s	**92.3**	**91.7**	**26.7**
Low - 3s	90.1	88.2	25.6
Low - 4s	88.2	84.7	23.3

Table 3. Results for ablation studies in terms of sequence length.

Models	Beats Coverage	Beats Hit	Genre Acc.
High w/o M	83.5	82.9	15.1
High w/o V	87.1	88.2	16.3
High (full)	**88.2**	**84.7**	**24.4**
Low w/o M	89.4	87.6	15.1
Low w/o V	90.6	90.0	17.4
Low (full)	**92.3**	**91.7**	**26.7**

Table 4. Results for ablation studies in terms of input modalities on the AIST++ dataset. *M* means the motion data, and *V* means the visual data.

Models	Beats Coverage	Beats Hit	Genre Acc.
High 1-layer D.	75.3	72.9	9.3
High 2-layer D.	85.3	82.9	21.0
High w/o scaling	72.9	71.8	14.0
High w/o reshape	73.5	70.1	11.6
High w/o fine-tune	87.0	**84.7**	**24.4**
High (full)	**88.2**	**84.7**	**24.4**
Low 1-layer D.	73.5	71.8	8.1
Low 2-layer D.	87.0	85.9	22.1
Low w/o scaling	72.4	70.1	12.8
Low w/o reshape	73.5	71.8	12.8
Low w/o fine-tune	**92.3**	91.2	**26.7**
Low (full)	**92.3**	**91.7**	**26.7**

Table 5. Results for ablation studies in terms of model architectures on the AIST++ dataset. *D.* means discriminators.

Data Modality. We perform ablation studies in terms of the input data modalities, by removing either the dance motion or the visual frame from the input data. Table 4 lists the corresponding experimental results. We observe that both motion and visual data contribute to our conditioned music generation task. Specifically, the motion data impose a larger impact on the musical rhythm, which is consistent with our expectations since the musical rhythm is closely correlated with the dance motion.

Model Architecture. We also test various variants of our *D2M-GAN* in terms of the model architecture and proposed model design techniques. The corresponding results are represented in Tabel 5. The experimental results show that the multi-scale layer for the discriminators, the scaling operation in the generator, as well as the reshape techniques for discriminators are crucial.

Losses	Beats Coverage	Beats Hit	Genre Acc.
High w/o L_{FM}	85.3	**84.7**	23.3
High w/o L_{wav}	85.9	**84.7**	23.3
High w/o L_{mel}	77.6	76.5	18.6
High (full)	**88.2**	**84.7**	**24.4**
Low w/o L_{FM}	91.7	90.1	24.4
Low w/o L_{wav}	89.4	88.8	23.3
Low w/o L_{mel}	78.8	77.1	17.4
Low (full)	**92.3**	**91.7**	**26.7**

Table 6. Results for ablation studies in terms of losses on the AIST++ dataset.

Loss function. We analyze the impact of different losses included in the overall training objective. The results from Table 6 show the contributions of each loss term. Specifically, we observe the audio perceptual loss from the frequency domain L_{mel} helps with the generation of musical rhythm, it is reasonable due to the fact that mel-spectrogram features help to capture the high frequencies from the audio signals, which is closely related to the dance beats.

5. Conclusion and Discussion

We propose *D2M-GAN* framework for complex music generation from dance videos via the VQ audio representations. Extensive experiments on multiple datasets, and comprehensive evaluations in terms of various musical characteristics prove the effectiveness of our method. We also introduce a novel TikTok dance-music dataset in this work.

As for limitations, the proposed *D2M-GAN* is an end-to-end framework, an interesting future direction is to explore how one can use controllable conditioning information to promote an interactively editing/generating system.

References

[1] Gunjan Aggarwal and Devi Parikh. Dance2music: Automatic dance-driven music generation. *arXiv preprint arXiv:2107.06252*, 2021. 1, 6, 7, 8

[2] Relja Arandjelovic and Andrew Zisserman. Look, listen and learn. In *ICCV*, 2017. 2

[3] Yusuf Aytar, Carl Vondrick, and Antonio Torralba. Soundnet: Learning sound representations from unlabeled video. *NeurIPS*, 2016. 2

[4] Jean-Pierre Briot, Gaëtan Hadjeres, and François-David Pachet. *Deep learning techniques for music generation*, volume 1. Springer, 2020. 1

[5] Z. Cao, G. Hidalgo Martinez, T. Simon, S. Wei, and Y. A. Sheikh. Openpose: Realtime multi-person 2d pose estimation using part affinity fields. *IEEE TPAMI*, 2019. 3

[6] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *CVPR*, 2017. 3

[7] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, 2017. 3

[8] Abe Davis and Maneesh Agrawala. Visual rhythm and beat. In *ACM Transactions on Graphics (TOG)*, 2018. 6

[9] Prafulla Dhariwal, Heewoo Jun, Christine Payne, Jong Wook Kim, Alec Radford, and Ilya Sutskever. Jukebox: A generative model for music. *arXiv preprint arXiv:2005.00341*, 2020. 1, 2, 3, 5, 6, 7, 8, 12

[10] Shangzhe Di, Zeren Jiang, Si Liu, Zhaokai Wang, Leyan Zhu, Zexin He, Hongming Liu, and Shuicheng Yan. Video background music generation with controllable music transformer. In *ACMMM*, 2021. 3, 6, 7, 8

[11] Chris Donahue, Julian McAuley, and Miller Puckette. Adversarial audio synthesis. In *ICLR*, 2019. 4

[12] Hao-Wen Dong, Wen-Yi Hsiao, Li-Chia Yang, and Yi-Hsuan Yang. Musegan: Multi-track sequential generative adversarial networks for symbolic music generation and accompaniment. In *AAAI*, 2018. 2, 3

[13] Daniel PW Ellis. Beat tracking by dynamic programming. *Journal of New Music Research*, 2007. 6

[14] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *CVPR*, 2021. 2, 3, 5

[15] Joao P 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. *Computers & Graphics*, 2021. 2

[16] Chuang Gan, Deng Huang, Peihao Chen, Joshua B Tenenbaum, and Antonio Torralba. Foley music: Learning to generate music from videos. In *ECCV*, 2020. 2, 3, 6, 7, 8

[17] Chuang Gan, Deng Huang, Hang Zhao, Joshua B Tenenbaum, and Antonio Torralba. Music gesture for visual sound separation. In *CVPR*, 2020. 2

[18] Ruohan Gao, Rogerio Feris, and Kristen Grauman. Learning to separate object sounds by watching unlabeled video. In *ECCV*, 2018. 2

[19] Ruohan Gao and Kristen Grauman. 2.5 d visual sound. In *CVPR*, 2019. 2

[20] Ruohan Gao and Kristen Grauman. Co-separating sounds of visual objects. In *ICCV*, 2019. 2

[21] Ruohan Gao, Tae-Hyun Oh, Kristen Grauman, and Lorenzo Torresani. Listen to look: Action recognition by previewing audio. In *CVPR*, 2020. 2

[22] Jort F Gemmeke, Daniel PW Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R Channing Moore, Manoj Plakal, and Marvin Ritter. Audio set: An ontology and human-labeled dataset for audio events. In *ICASSP*. IEEE, 2017. 7

[23] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NIPS*, 2014. 3

[24] Shawn Hershey, Sourish Chaudhuri, Daniel PW Ellis, Jort F Gemmeke, Aren Jansen, R Channing Moore, Manoj Plakal, Devin Platt, Rif A Saurous, Bryan Seybold, et al. Cnn architectures for large-scale audio classification. In *ICASSP*. IEEE, 2017. 7

[25] Cheng-Zhi Anna Huang, Ashish Vaswani, Jakob Uszkoreit, Noam Shazeer, Ian Simon, Curtis Hawthorne, Andrew M Dai, Matthew D Hoffman, Monica Dinculescu, and Douglas Eck. Music transformer: Generating music with long-term structure. In *ICLR*, 2019. 2, 3

[26] Vladimir Iashin and Esa Rahtu. Taming visually guided sound generation. In *British Machine Vision Conference (BMVC)*, 2021. 3

[27] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *CVPR*, 2017. 5

[28] Shulei Ji, Jing Luo, and Xinyu Yang. A comprehensive survey on deep music generation: Multi-level representations, algorithms, evaluations, and future directions. *arXiv preprint arXiv:2011.06801*, 2020. 1

[29] Hsuan-Kai Kao and Li Su. Temporally guided music-to-body-movement generation. In *ACMMM*, 2020. 2

[30] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017. 3

[31] Evangelos Kazakos, Arsha Nagrani, Andrew Zisserman, and Dima Damen. Epic-fusion: Audio-visual temporal binding for egocentric action recognition. In *ICCV*, 2019. 2

[32] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *ICLR*, 2014. 3

[33] Jungil Kong, Jaehyeon Kim, and Jaekyoung Bae. Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis. In *NIPS*, 2020. 4, 5

[34] Bruno Korbar, Du Tran, and Lorenzo Torresani. Cooperative learning of audio and video models from self-supervised synchronization. In *NeurIPS*, 2018. 2

[35] Kundan Kumar, Rithesh Kumar, Thibault de Boissiere, Lucas Gestin, Wei Zhen Teoh, Jose Sotelo, Alexandre de Brébisson, Yoshua Bengio, and Aaron C Courville. Melgan: Generative adversarial networks for conditional waveform synthesis. In *NIPS*, 2019. 3, 4, 5

[36] Anders Boesen Lindbo Larsen, Søren Kaae Sønderby, Hugo Larochelle, and Ole Winther. Autoencoding beyond pixels using a learned similarity metric. In *International conference on machine learning*. PMLR, 2016. 5

[37] Hsin-Ying Lee, Xiaodong Yang, Ming-Yu Liu, Ting-Chun Wang, Yu-Ding Lu, Ming-Hsuan Yang, and Jan Kautz. Dancing to music. In *NIPS*, 2019. 1, 2, 6, 13

[38] Buyu Li, Yongchi Zhao, and Lu Sheng. Dancenet3d: Music based dance generation with parametric motion transformer. *arXiv preprint arXiv:2103.10206*, 2021. 1, 2

[39] Ruilong Li, Shan Yang, David A. Ross, and Angjoo Kanazawa. Ai choreographer: Music conditioned 3d dance generation with aist++. In *ICCV*, 2021. 1, 2, 5, 7, 8, 13

[40] Jae Hyun Lim and Jong Chul Ye. Geometric gan. *arXiv preprint arXiv:1705.02894*, 2017. 5

[41] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: A skinned multi-person linear model. *ACM Trans. Graphics (Proc. SIGGRAPH Asia)*, 34, 2015. 3

[42] Mario Lucic, Karol Kurach, Marcin Michalski, Sylvain Gelly, and Olivier Bousquet. Are gans created equal? a large-scale study. In *NeurIPS*, 2018. 4

[43] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. *arXiv preprint:1802.05957*, 2018. 5

[44] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. In *ICLR*, 2016. 3

[45] Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In *NIPS*, 2017. 2, 3, 5, 12

[46] Sageev Oore, Ian Simon, Sander Dieleman, Douglas Eck, and Karen Simonyan. This time with feeling: Learning expressive musical performance. *Neural Computing and Applications*, pages 955–967, 2020. 2, 3

[47] Andrew Owens and Alexei A Efros. Audio-visual scene analysis with self-supervised multisensory features. In *ECCV*, 2018. 2

[48] Andrew Owens, Jiajun Wu, Josh H McDermott, William T Freeman, and Antonio Torralba. Ambient sound provides supervision for visual learning. In *ECCV*, 2016. 2

[49] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. dÁlché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. 12

[50] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015. 3, 4

[51] Tanzila Rahman, Bicheng Xu, and Leonid Sigal. Watch, listen and tell: Multi-modal weakly supervised dense event captioning. In *ICCV*, 2019. 2

[52] Ali Razavi, Aaron van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with vq-vae-2. In *NIPS*, 2019. 2, 3, 5

[53] Xuanchi Ren, Haoran Li, Zijian Huang, and Qifeng Chen. Self-supervised dance video synthesis conditioned on music. In *ACM MM*, 2020. 1, 2

[54] Tim Sainburg, Marvin Thielk, and Timothy Q Gentner. Finding, visualizing, and quantifying latent structure across diverse animal vocal repertoires. *PLoS computational biology*, 16(10):e1008228, 2020. 6

[55] Tim Salimans and Durk P Kingma. Weight normalization: A simple reparameterization to accelerate training of deep neural networks. In *NeurIPS*, 2016. 4

[56] Eli Shlizerman, Lucio Dery, Hayden Schoen, and Ira Kemelmacher-Shlizerman. Audio to body dynamics. In *CVPR*, 2018. 1, 2

[57] Kun Su, Xiulong Liu, and Eli Shlizerman. Audeo: Audio generation for a silent performance video. In *NeurIPS*, 2020. 2

[58] Taoran Tang, Jia Jia, and Hanyang Mao. Dance with melody: An lstm-autoencoder approach to music-oriented dance synthesis. In *ACMMM*, 2018. 2

[59] Yapeng Tian, Jing Shi, Bochen Li, Zhiyao Duan, and Chenliang Xu. Audio-visual event localization in unconstrained videos. In *ECCV*, 2018. 2

[60] Shuhei Tsuchida, Satoru Fukayama, Masahiro Hamasaki, and Masataka Goto. Aist dance video database: Multi-genre, multi-dancer, and multi-camera database for dance information processing. In *Proceedings of the 20th International Society for Music Information Retrieval Conference, (ISMIR)*, 2019. 2, 5, 6, 13

[61] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *CVPR*, 2018. 4

[62] Xin Wang, Yuan-Fang Wang, and William Yang Wang. Watch, listen, and describe: Globally and locally aligned cross-modal attentions for video captioning. In *NAACL*, 2018. 2

[63] Yu Wu and Yi Yang. Exploring heterogeneous clues for weakly-supervised audio-visual video parsing. In *CVPR*, 2021. 2

[64] Yu Wu, Linchao Zhu, Yan Yan, and Yi Yang. Dual attention matching for audio-visual event localization. In *ICCV*, 2019. 2

[65] Bing Xu, Naiyan Wang, Tianqi Chen, and Mu Li. Empirical evaluation of rectified activations in convolutional network. *arXiv preprint arXiv:1505.00853*, 2015. 3

[66] Hang Zhao, Chuang Gan, Wei-Chiu Ma, and Antonio Torralba. The sound of motions. In *ICCV*, 2019. 2, 3

[67] Ye Zhu, Yu Wu, Hugo Latapie, Yi Yang, and Yan Yan. Learning audio-visual correlations from variational cross-modal generation. In *ICCASP*, 2021. 2

[68] Wenlin Zhuang, Congyi Wang, Siyu Xia, Jinxiang Chai, and Yangang Wang. Music2dance: Dancenet for music-driven dance generation. *arXiv preprint arXiv:2002.03761*, 2020. 2

A. Network Architecture

A.1. Generator

The following Table 7, Table 8, Table 9 and Table 10 show the detailed model architectures of the motion encoder, high-level VQ generator, low-level VQ generator and the residual block, respectively.

6 × 1, stride=1, Conv 256, LeakyReLU
Residual Stack 256
3 × 1, stride=1, Conv 512, LeakyReLU
Residual Stack 512
3 × 1, stride=1, Conv 1024, LeakyReLU
Residual Stack 1024
3 × 1 , stride=1, Conv 1024, LeakyReLU
4 × 1, stride=1, Conv 1

Table 7. Architecture for the motion encoder.

6 × 1, stride=2, Conv 32, LeaklyReLU
Residual Stack 32
41 × 1, stride=2, Conv 64, LeakyReLU
Residual Stack 64
41 × 1, stride=1, Conv 128, LeakyReLU
Residual Stack 128
41 × 1, stride=1, Conv 256, LeaklyReLU
Residual Stack 256
41 × 1, stride=1, Conv 512, LeakyReLU
Residual Stack 512
40 × 1, stride=1, Conv 64
Tanh()

Table 8. Architecture for the high-level VQ generator.

6 × 1, stride=2, Conv 32, LeakyReLU
Residual Stack 32
4 × 1, stride=1, Conv 64, LeakyReLU
Residual Stack 64
40 × 1, stride=2, Conv 128, LeakyReLU
Residual Stack 128
40 × 1, stride=1, Conv 256, LeakyReLU
Residual Stack 256
40 × 1, stride=1, Conv 512, LeakyReLU
Residual Stack 512
40 × 1, stride=1, Conv 1024, LeakyReLU
Residual Stack 1024
40 × 1, stride=1, Conv 1024, LeakyReLU
40 × 1, stride=1, Conv 64, LeakyReLU
Tanh()

Table 9. Architecture for the low-level VQ generator.

LeakyReLU, dilation=1, Conv
LeakyReLU, dilation=1. Conv
Shortcut Path
LeakyReLU, dilation=3, Conv
LeakyReLU, dilation=1, Conv
Shortcut Path
LeakyReLU, dilation=9, Conv
LeakyReLU, dilation=1, Conv
Shortcut Path

Table 10. Architecture for the residual stack.

A.2. Discriminator

We adopt the multi-scale discriminator design for the proposed *D2M-GAN*, where is formed by a stack of 3 discriminator blocks that operates on the original VQ sequence, and its downsampled features based on the window-based objective functions as introduced in the main paper. The architecture of each discriminator block is shown below in Table 11.

15 × 1, stride=1, Conv 16, LeakyReLU
41 × 1, stride=4, Groups=4, Conv 64, LeakyReLU
41 × 1, stride=4, Groups=16, Conv 256, LeakyReLU
41 × 1, stride=4, Groups=64, Conv 1024, LeakyReLU
41 × 1, stride=4, Groups=256, Conv 1024, LeakyReLU
5 × 1, stride=1, Conv 1024, LeakyReLU
3 × 1, stride=1, Conv 1

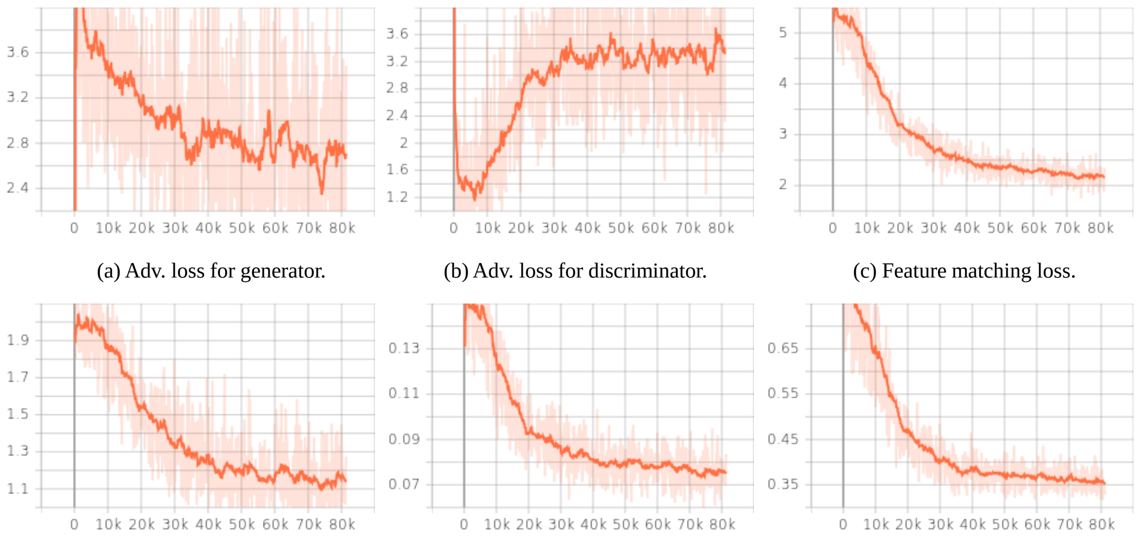
Table 11. Architecture for the discriminator block.

B. Experimental Details

We implement the entire framework using the PyTorch [49] framework for automatic differentiation and GPU-accelerated training and inference.

Pre-learned Codebook. We adopt two independently pre-trained codebooks for two levels in our *D2M-GAN*. Specifically, the original JukeBox [9] contains three levels of VQ-VAE [45] based models, which are defined as top, middle and bottom levels with hop lengths of 128, 32, and 8, respectively. We adopt the top level codebook for the high-level *D2M-GAN*, and the middle level codebook for the low-level *D2M-GAN*. Therefore, for a two-second audio sequence with a sampling rate of 22050 Hz, the generated VQ sequences from the high-level and low-level VQ generators are in dimension of 64 × 344 and 64 × 1378, respectively, where 64 is the dimension of the codebook entry, 344 and 1378 are the sequence lengths.

Training Losses. Since our proposed *D2M-GAN* includes



(a) Adv. loss for generator.

(b) Adv. loss for discriminator.

(c) Feature matching loss.

(d) Codebook commitment loss.

(e) Audio waveform loss.

(f) Audio mel-spectrogram loss.

Figure 6. **Training losses for the proposed *D2M-GAN*.** *Adv.* stands for adversarial.

multiple loss terms in the overall training objective, we show the change of each loss term during the training process in Figure 6. It is worth noting the model architectures and techniques described in our main paper are crucial for *D2M-GAN* to maintain a stable training. Notably, the codebook commitement loss, audio waveform loss and audio mel-spectrogram loss can reach the comparable levels with the GT audio samples after convergence.

C. TikTok Dance-Music Dataset

The current version of our TikTok dance-music dataset contains in total 445 videos, which we annotate from 15 TikTok dance video compilations. There are 85 different songs, with majority of videos having a single dance performer, and a maximum of five performers. The average length of each video is approximately 12.5s. We split the training and testing set based on the music IDs, and ensure that there are no overlapping songs for two splits.

Compared to the existing music and dance datasets such as AIST++ [39, 60], our dataset is closer to the real-world scenario with various background, which is also our initial motivation to introduce this dataset. Additionally, majority of the current datasets available are not initially proposed for the dance to music generation task, AIST [60] is designed for dance music processing, AIST++ [39] provides the extra annoations for the subset of AIST for generating dance motion conditioned on music, some other similar datasets for motion generation have also been introduced [37]. Therefore, we hope that our proposed TikTok dance-music dataset can serve as a starting point for relevant future researches.

D. Subjective Evaluations

We conduct the Mean Opinion Scores (MOS) test for the subjective evaluations. In total, 26 subjects participated our MOS tests, among which 9 of them are females, the rest are males.

Two of our music evaluation protocols are based on the human subjective evaluations, which are the dance-music coherence test and the music overall quality test. For the dance-music coherence test, each evaluator is asked to rate 15 dance videos that are post-processed by fusing the original visual frames and generated music samples from different models. Specifically, the evaluators are asked to rate from the coherence aspect of the dance video (*i.e.*, whether they feels the music is coherent with the dance moves) with reference to the GT videos and original music. For the overall quality test, 15 audio samples (without video frames) are played during the test for each evaluator, after which the evaluator is asked to rate the sound quality from the score range of 1 to 5. It is worth noting that for the overall quality test, we do not compare with the music samples obtained from the symbolic MIDI representation based methods. This is due to the reason that the symbolic representations and pre-defined music synthesizers in nature do not introduce audio noises to the generated signals, which makes the music samples sound rather “clean and high-quality”, while the continuous or VQ audio representations can hardly achieve the similar effects with a learned music synthesizer (samples included in our demo video). Therefore, we do not include the MIDI-based methods as our baselines for fairness considerations.