2022 IEEE International Conference on Robotics and Automation (ICRA)
May 23-27, 2022. Philadelphia, PA, USA

2022 IEEE International Conference on Robotics and Automation (ICRA) | 978-1-7281-9681-7/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICRA46639.2022.9811649

DanceHAT: Generate Stable Dances for Humanoid Robots with Adversarial Training

Buqing Nie[1] and Yue Gao[2]

Abstract— **Music to dance for humanoid robots is an interesting task. Robot dance generation is challenging when considering music pieces, human dancer motions, and robot stability simultaneously. Previous methods rely on human-designed motion library or stability constraints for robot postures. Hence, dance generation for humanoid robots requires expert design, which can be time-consuming across different humanoid platforms. In this work, we propose a novel method called DanceHAT, which generates stable humanoid dances by imitating human dancers with self-learning. DanceHAT is an adversarial training framework, which incorporates similarity loss and stability loss simultaneously. Furthermore, DanceHAT does not require human-designed features or robot model information. Experiments in the simulation environment and on the real robot demonstrate that our model can generate stable, diverse, and human-like dances for humanoid robots automatically. In addition, DanceHAT is a general training approach for robot imitation tasks with stability constraints, thus can be utilized in other humanoid tasks and will be researched in future works.**

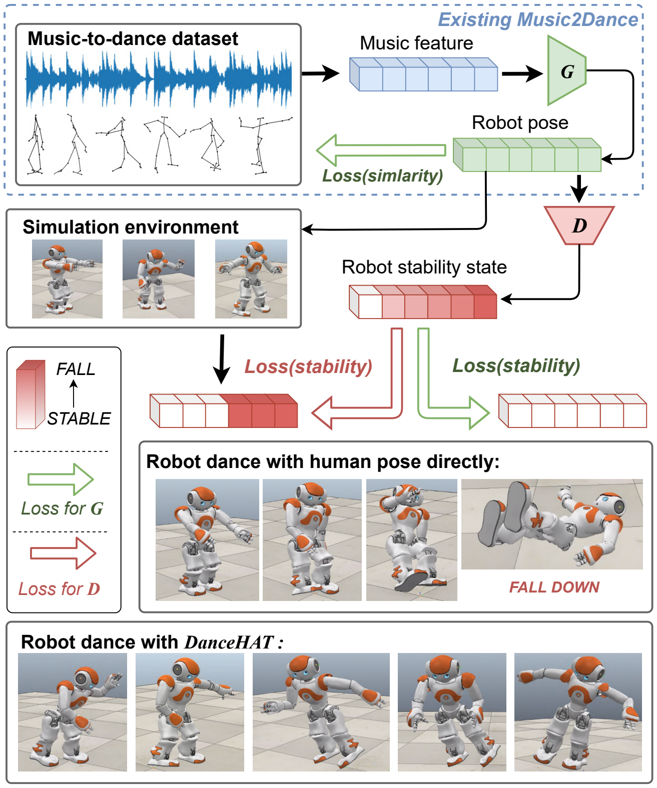

Fig. 1: The model architecture of DanceHAT. DanceHAT incorporates similarity loss and stability constraints through adversarial training. It can imitate human dances while satisfying robot stability constraints.

I. INTRODUCTION

Dance is an important aesthetic art in human culture, which plays an essential role in daily entertainment and social interactions. Recently, generating robot dances, especially generating humanoid robot dances has become an active research topic [1]–[5]. The humanoid robots have human-like structure and human-friendly appearances, hence more popular than other legged robots in dance scenarios including dance teaching [2] and human dance imitation [4]. However, designing dances for humanoid robots manually requires extensive experience and art talents, thus is quite challenging. Besides, tuning dance motions to satisfy robot dynamics constraints such as stability requires expert knowledge and is also time-consuming, which limits dance applications of humanoid robots in our daily lives.

Over the past two decades, generating humanoid robot dances automatically has been actively researched. One major approach for this task is generating robot dances utilizing music features directly without reference motions [3], [4], [6]–[8]. Some excellent works are proposed utilizing various methods, including Markov model [4], [6], key frame-based designer [8] and task-based designer [7]. These methods can synthesize flexible and stable dances for a given robot based on the music input. However, extensive expert knowledge is still necessary for these methods, such as designing robot motion library and maintaining robot stability, which is time-consuming and complicated in practice.

Another approach to synthesizing robot dances is to imitate human dances [1], [9]–[11]. Plenty of excellent works are proposed with various methods, such as pre-defined robot action modes [11] and generative auto-regressive model [1]. With the development of deep learning, more interesting methods are developed utilizing various deep learning models, including LSTM [10], VAE [12], GAN [13], and transformer [14]. However, as illustrated in Fig. 1, existing works mainly focus on the similarity between the synthesized dance and the corresponding human dance, which do not incorporate the robot stability in the model architecture. Thus, few previous works have been applied to the real

This work is supported by Ministry of Science and Technology of the People's Republic of China (Grant No. 2021YFF0306202), National Natural Science Foundation of China (Grant No. 61903247), and Shanghai Municipal Science and Technology Major Project (Grant No. 2021SHZDZX0102).

[1]Buqing Nie is with Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, 200240, P.R. China niebuqing@sjtu.edu.cn

[2]Yue Gao is with MoE Key Lab of Artificial Intelligence and AI Institute of Shanghai Jiao Tong University, Shanghai, 200240, P.R. China. yuegao@sjtu.edu.cn

humanoid robots without manual constraints on the poses.

In this paper, we propose a new method to generate humanoid robot dances called **DanceHAT** (Generate Stable **Dance**s for **H**umanoid Robots with **A**dversarial **T**raining). Our method considers robot stability and similarity to human dances simultaneously without any expert knowledge on the robot dynamics. As shown in Fig. 1, DanceHAT is designed based on the adversarial training framework composed of a generator G and a discriminator D. During training, G is utilized to imitate human dances and adjust robot motions for stability through the similarity loss and stability loss accordingly. D is considered as a classifier and trained to predict the robot stability state during dancing. Experiments on the humanoid robot NAO [15] and ROBOTIS OP2 [16] demonstrate that DanceHAT can generate stable dances for humanoid robots by imitation without pre-defined motion library or human-designed constraints on robot motions.

The contributions of this paper can be summarized as follows:

1) DanceHAT can compute a balanced solution which minimizes the difference between generated robot dances and target human dances, meanwhile maintaining robot stability. Thus, without human-designed motion library or expert knowledge on robot stability, it can learn to generate stable dances across different humanoid robot platforms.
2) DanceHAT is a general adversarial training framework for imitation tasks among humanoid robots. To our knowledge, this is the first study to maintain robot stability based on adversarial training in humanoid robot imitation tasks.
3) Experiments on the humanoid robot NAO and ROBOTIS OP2 are conducted. The results in the simulation environment and on the real robot demonstrate that DanceHAT can effectively generate stable dances for humanoid robots according to different music pieces.

II. RELATED WORK

A. Music to Dance Synthesis for Human

Recently, human dance synthesis has been studied actively. Tang et al. [10] proposed an LSTM-autoencoder model to synthesize 3D human dance motions sensitive to the music genres and emotions. Lee et al. [12] proposed a novel framework to dismantle and assemble between complex dances and basic dance units. Guo et al. [17] proposed a novel framework to generate natural and diverse human dances utilizing Lie Algebra based VAE model. Li et al. [14] proposed Full Attention Cross-Modal Transformer model (FACT) for long motion sequence generation correlated with the music. Ren et al. [18] proposed a self-supervised model to synthesize realistic and diverse dance videos. Huang et al. [19] designed a novel seq2seq architecture for long-term dance generation with curriculum learning. Although these excellent methods can synthesize diverse human dances easily, the dances generated are unsuitable for humanoid robots because of various constraints such as stability problems.

B. Music to Dance Synthesis for Robot

Robot dance generation is a classic and active research topic. Kojima et al. [4] designed a humanoid robot dance system composed of a foot states classifier and a key-frame detector. Nakaoka et al. [9], [11] developed a dance imitation method based on key-states and human designed tasks. Recently, various deep learning methods have been developed [10], [12], [13]. Hyemin et al. [1] proposed a deep learning based framework, which can synthesize diverse dances for humanoid robots. However, robot stability is not considered in some recent works, while other works ensure stability by designing motion library [8] or designing geometric constraints based on ZMP [20], which require expert knowledge on robot dynamics. In this work, a new model for humanoid dance synthesis is proposed based on adversarial training, which can generate stable dances automatically across different humanoid robots.

C. Generative Adversarial Networks

Generative Adversarial Networks (GAN) [21] is a classic and important framework for generative tasks. It has shown considerable potential in various tasks since proposed by [21], including image generation [22], image translation [23], and domain adaptation [24]. Recently, GAN has been widely used in robot motion generation [13], [25]. However, existing works mainly focus on the similarity of synthesized motions with corresponding target motions, while the robot stability is not taken into consideration. In this work, we propose a new method DanceHAT to incorporate similarity and stability based on adversarial training.

III. METHODOLOGY

A. Problem Formulation

In this section, we formulate the robot dance synthesis task with mathematical descriptions. As shown in Fig. 2, a human dance dataset and a simulation environment for humanoid robot $\mathcal{R}$ is given. The human dance dataset $(M, P^{\mathcal{H}})$ is composed of acoustic features of N pieces of music $M = \{m_i | 1 \leq i \leq N\}$ and corresponding human dances $P^{\mathcal{H}} = \{p_i^{\mathcal{H}} | 1 \leq i \leq N\}$, where m_i, $p_i^{\mathcal{H}}$ are the music feature and human pose feature of the i-th piece of music respectively. T_i denotes the time length of the i-th piece of music and dance. Given a humanoid robot $\mathcal{R}$ with specific dynamics, $P^{\mathcal{R}}$ denotes the space of possible robot dances. $\forall p_i^{\mathcal{R}} \in P^{\mathcal{R}}$ $(1 \leq i \leq N)$ denotes the robot dance synthesized for the i-th piece of music m_i[1].

In order to measure the stability of the robot $\mathcal{R}$ during dancing, we define a function $\mathcal{F}(p_i^{\mathcal{R}}) \in [0,1]^{T_i}$, where $\mathcal{F}(p_i^{\mathcal{R}})_t$ denotes the stability state of the robot $\mathcal{R}$ dancing with motion sequence $p_i^{\mathcal{R}}$ at time t, i.e.

$$\mathcal{F}(p_i^{\mathcal{R}})_t = \begin{cases} 0 & \text{if robot } \mathcal{R} \text{ is stable at time } t; \\ 1 & \text{if robot } \mathcal{R} \text{ has fell down at time } t. \end{cases} \quad (1)$$

[1] The music feature m_i, robot pose feature $p_i^{\mathcal{R}}$, and human pose feature $p_i^{\mathcal{H}}$ are two dimensional matrices, which are supposed to be in bold style. Normal style is utilized in this work for convenience and readability.

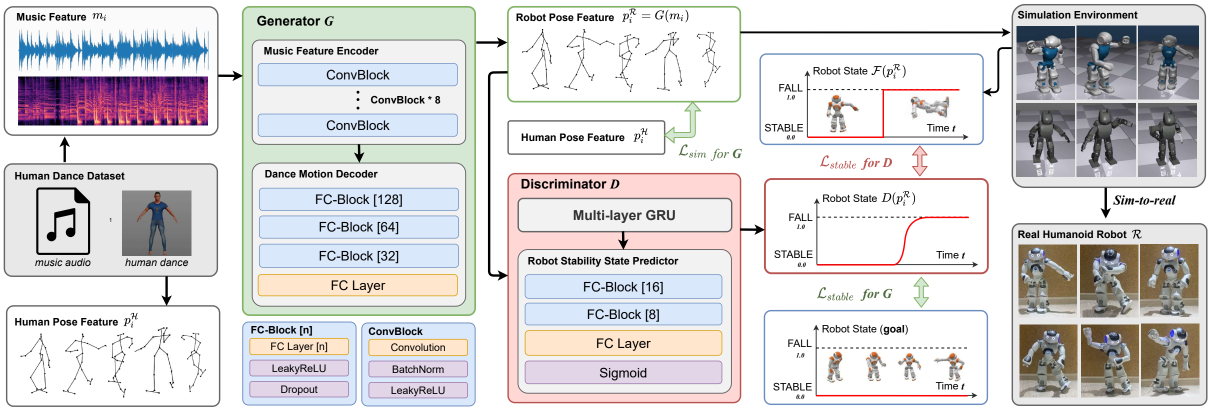

Fig. 2: The detailed model architecture of DanceHAT. DanceHAT is designed based on the adversarial training framework and composed of two deep neural networks G and D. The discriminator D predicts robot stability during dancing. The generator G synthesizes robot dances based on musics input, which is trained to imitate human dances and adjust poses as slightly as possible to maintain robot stability. More details are shown in the experiment section and the video attached.

We can express $\mathcal{F}(p_i^{\mathcal{R}})_t$ in the form of a special step function on the time axis, i.e.

$$\mathcal{F}(p_i^{\mathcal{R}})_t = \mathbb{1}(t - t_f) = \begin{cases} 0 & \text{if } t < t_f; \\ 1 & \text{if } t \geq t_f, \end{cases} \tag{2}$$

where $\mathbb{1}(t)$ denotes the unit step function and t_f is defined as the time when the robot $\mathcal{R}$ falls down. As described above, the ground truth of stability $\mathcal{F}(p_i^{\mathcal{R}})_t$ is a discrete value, which can be obtained from simulation quite easily.

Our goal is to construct a generator $G(m_i)$ for humanoid robot dance synthesis based on the music, i.e. $G(m_i) \to p_i^{\mathcal{R}}$. The generated dance is as similar to the corresponding human dance $p_i^{\mathcal{H}}$ as possible while the robot $\mathcal{R}$ maintains stable during dancing. In other words, G needs to find the balance between the similarity goal and the robot stability constraint. The mathematical description for this task is as follows:

$$\begin{aligned} \min_G \quad & \mathbb{E}_{m_i, p_i^{\mathcal{H}} \sim (M, P^{\mathcal{H}})} \left[\mathcal{L}_{sim} \left(G(m_i), p_i^{\mathcal{H}} \right) \right] \\ s.t. \quad & \mathcal{F}(G(m_i)) = 0, \forall m_i \in M, \end{aligned} \tag{3}$$

where $\mathcal{L}_{sim}$ is the similarity loss and measures the similarity between two input dances. In the following sections, N_m denotes the dimension of the music feature m_i; $N_{\mathcal{H}}$ and $N_{\mathcal{R}}$ denote the dimension of the human pose feature $p_i^{\mathcal{H}}$ and robot pose feature $p_i^{\mathcal{R}}$ respectively.

B. Generate Stable Dances with Adversarial Training

In this section, we transform the problem Eq. (3) into the problem form proposed in [21], which can be solved by adversarial training. Based on Eq. (3), we can obtain the following optimization problem:

$$\begin{aligned} \min_G \quad & \mathbb{E}_{m_i, p_i^{\mathcal{H}} \sim (M, P^{\mathcal{H}})} \left[\mathcal{L}_{sim} \left(G(m_i), p_i^{\mathcal{H}} \right) \right. \\ & \left. + \lambda \mathcal{L}_{stable} \left(\mathcal{F}(G(m_i)), 0 \right) \right], \end{aligned} \tag{4}$$

where λ is a Lagrange multiplier and $\mathcal{L}_{stable}$ denotes stability loss. We assume $\mathcal{L}_{stable} \left(\mathcal{F} \left(p_i^{\mathcal{R}} \right), 0 \right) \geq 0$ for $\forall p_i^{\mathcal{R}} \in P^{\mathcal{R}}$, and $\mathcal{L}_{stable} \left(\mathcal{F} \left(p_i^{\mathcal{R}} \right), 0 \right) = 0 \Leftrightarrow \mathcal{F} \left(p_i^{\mathcal{R}} \right) = 0$, i.e. the robot $\mathcal{R}$ remains stable consistently. $\mathcal{L}_{stable}$ satisfying this assumption can be designed as Binary Cross Entropy (BCE) Loss. Eq. (3) and Eq. (4) are equivalent when $\lambda \to +\infty$.

However, the formula of the function $\mathcal{F}$ is unknown thus indifferentiable, which prevents us from solving Eq. (4) utilizing gradient descent (GD) based algorithms. Thus, a differentiable discriminator D is proposed to replace $\mathcal{F}$, i.e.

$$\begin{aligned} \min_G \quad & \mathbb{E}_{m_i, p_i^{\mathcal{H}} \sim (M, P^{\mathcal{H}})} \left[\mathcal{L}_{sim} \left(G(m_i), p_i^{\mathcal{H}} \right) \right. \\ & \left. + \lambda \mathcal{L}_{stable} \left(D(G(m_i)), 0 \right) \right] \\ s.t. \quad & D(p_i^{\mathcal{R}}) = \mathcal{F}(p_i^{\mathcal{R}}), \ \forall p_i^{\mathcal{R}} \in \{ G(m_i) \mid m_i \in M \}, \end{aligned} \tag{5}$$

where $D(p_i^{\mathcal{R}})_t \in [0, 1]$ denotes the probability of the robot $\mathcal{R}$ falling at time t predicted by D, thus is a continuous value. Then the following equivalent optimization problem Eq. (6) is obtained:

$$\begin{aligned} \min_G \ & \mathbb{E}_{m_i, p_i^{\mathcal{H}} \sim (M, P^{\mathcal{H}})} \left[\mathcal{L}_{sim} \left(G(m_i), p_i^{\mathcal{H}} \right) \right. \\ & \left. + \lambda \mathcal{L}_{stable} \left(D(G(m_i)), 0 \right) \right] \\ + \ & \min_D \ \tau \, \mathbb{E}_{m_i \sim M} \left[\mathcal{L}_{stable} \left(D(G(m_i)), \mathcal{F}(G(m_i)) \right) \right], \end{aligned} \tag{6}$$

where λ and τ are Lagrange multipliers with $\lambda \to +\infty$, $\tau \to +\infty$. The problem Eq. (6) is quite similar to the problem proposed in [21], thus could be solved by the adversarial training method.

In this work, a new method called DanceHAT for humanoid dance synthesis is proposed based on Eq. (6) and adversarial training. DanceHAT is composed of a dance generator G and a robot stability state discriminator D. Loss functions for G and D are illustrated in Eq. (7) and Eq. (8) respectively:

$$\begin{aligned} \mathcal{L}_G = \alpha \, & \mathbb{E}_{m_i, p_i^{\mathcal{H}} \sim (M, P^{\mathcal{H}})} \left[\mathcal{L}_{sim} \left(G(m_i), p_i^{\mathcal{H}} \right) \right] \\ & + \beta \, \mathbb{E}_{m_i \sim M} \left[\mathcal{L}_{stable} \left(D(G(m_i)), 0 \right) \right], \end{aligned} \tag{7}$$

$$\mathcal{L}_D = \mathbb{E}_{m_i \sim M} \left[\mathcal{L}_{stable} \left(D(G(m_i)), \mathcal{F}(G(m_i)) \right) \right], \tag{8}$$

where α and β are weights for the loss functions.

C. Model Framework

As illustrated in Eq. (6) and Fig. 2, DanceHAT is composed of a generator G and a discriminator D. The following is one possible architecture utilized in this work.

1) Generator G: As shown in Fig. 2, the generator G is composed of a music encoder and a pose sequence decoder. The music feature encoder is a convolutional neural network, which extracts important information from the input music features. The dance motion decoder synthesize robot dances $p_i^{\mathcal{R}}$ based on music information extracted by the encoder.

As illustrated in Eq. (7), $\mathcal{L}_{sim}$ measures the similarity between synthesized dances and human dances. $\mathcal{L}_{sim}$ can be designed as Mean Square Error (MSE), if robot pose feature $p_i^{\mathcal{R}}$ and human pose feature $p_i^{\mathcal{H}}$ have same dimensions, i.e. $N_{\mathcal{R}} = N_{\mathcal{H}}$. Otherwise, other differentiable $\mathcal{L}_{sim}$ could be constructed, according to definitions of pose features. In this work, we choose MSE loss for $\mathcal{L}_{sim}$ and BCE Loss for $\mathcal{L}_{stable}$. The loss function for G is illustrated as Eq. (9):

$$\begin{aligned}\mathcal{L}_G = {} & \mathbb{E}_{m_i, p_i^{\mathcal{H}} \sim (M, P^{\mathcal{H}})} \left[\frac{1}{N_{\mathcal{H}} \times T_i} \sum_t^{T_i} \alpha_t \sum_j^{N_{\mathcal{H}}} \left(G\left(m^i\right)_{t,j} - p^{\mathcal{H}}_{i,t,j} \right)^2 \right] \\ & - \mathbb{E}_{m_i \sim M} \left[\frac{1}{T_i} \sum_t^{T_i} \beta_t \log\left(1 - D\left(G\left(m_i\right)\right)_t\right) \right],\end{aligned} \tag{9}$$

where $p^{\mathcal{H}}_{i,t,j}$ is the j-th pose feature value of the pose sequence $p_i^{\mathcal{H}}$ at time t. T_i denotes the time length of the i-th piece of music m_i. $\alpha \in \mathbb{R}^{T_i}$ and $\beta \in \mathbb{R}^{T_i}$ are hyper-parameters.

2) Discriminator D: The discriminator D is composed of a Gated Recurrent Unit (GRU) [26] and a robot stability state predictor. Given robot dances $p_i^{\mathcal{R}}$, the GRU extracts pose features related to the robot stability. Afterward, the stability state predictor predicts probability of the robot falling down during dancing, i.e. $D(p_i^{\mathcal{R}})_t \in [0,1]$.

As illustrated in Eq. (8), D is designed to predict robot stability, i.e. fit $\mathcal{F}$ with the stability loss $\mathcal{L}_{stable}$. During training, the ground truth of robot stability, i.e. $\mathcal{F}(p_i^{\mathcal{R}})$ is obtained by performing robot dances in the simulation environment. BCE loss is utilized for $\mathcal{L}_{stable}$ in this work. Thus, we can obtain the following loss function for D:

$$\begin{aligned}\mathcal{L}_D = {} & \mathbb{E}_{p_i^{\mathcal{R}} \sim \{G(m_i)\}} \left[\frac{1}{T_i} \sum_{t=1}^{T_i} \left[-\mathcal{F}(p_i^{\mathcal{R}})_t \log\left(D\left(p_i^{\mathcal{R}}\right)_t\right) \right.\right. \\ & \left.\left. - \left(1 - \mathcal{F}(p_i^{\mathcal{R}})_t\right) \log\left(1 - D\left(p_i^{\mathcal{R}}\right)_t\right) \right] \right] \\ = {} & \mathbb{E}_{p_i^{\mathcal{R}} \sim \{G(m_i)\}} \left[-\frac{1}{T_i} \left[\sum_{t=1}^{t_f - 1} \left(\log\left(1 - D\left(p_i^{\mathcal{R}}\right)_t\right)\right) \right.\right. \\ & \left.\left. - \sum_{t=t_f}^{T_i} \log\left(D\left(p_i^{\mathcal{R}}\right)_t\right) \right] \right] \qquad \text{(see Eq. (2))},\end{aligned} \tag{10}$$

where t_f is the time instance when robot $\mathcal{R}$ falls down.

3) Training Procedure: DanceHAT is trained with mini-batch stochastic gradient descent. In each iteration, given batches of musics m_i, G synthesizes corresponding robot dances, i.e. $p_i^{\mathcal{R}} = G(m_i)$, and D predicts robot stability, i.e. $D(p_i^{\mathcal{R}})$ through forward propagation. The ground truth of stability states $\mathcal{F}\left(p_i^{\mathcal{R}}\right)$ are obtained from simulation. Afterward, G and D are trained through Eq. (9) and Eq. (10) accordingly. After training for K iterations, the synthesized dances are as similar as possible to the human dances, under the conditions of robot stability, which is the goal proposed in the problem formulation as illustrated in Eq. (3).

IV. EXPERIMENTS

A. Experimental Setup

In this work, the humanoid robot NAO (Softbank, Tokyo, Japan) [15] and ROBOTIS OP2 (ROBOTIS, Seoul, Korea) [16] are utilized to conduct experiments. NAO is an compact and flexible humanoid robot, which has 25 degree of freedom (DOF) and 21 DOF are utilized in this work. The joints on wrists and hands are ignored, since they have no influence on the visual effects or robot stability. ROBOTIS OP2 is a high performance humanoid robot with 20 DOF. In the training procedure, we utilize Mujoco [27] as the simulation environment.

B. Dataset and Preprocessing

1) Human Dance Dataset: Human dance dataset proposed in [10] is utilized, which provides 61 music audios with corresponding 3D human dances. The music audios are provided in MP3 format with time lengths ranging from 50 seconds to 2 minutes. The 3D human dances are composed of 3D skeleton frames of human dancers, represented by the 3D coordinates of 21 key body points.

This human dance dataset contains four genres: Rumba, Cha-cha, Tango, and Waltz. However, Tango and Waltz dances in the dataset are duet dances, which are mainly composed of dramatic position movements and delicate poses of the lower body, such as fast rotation of the whole body. These motions are out of NAO's and ROBOTIS OP2's movement capability, thus unsuitable for this experiment. Therefore, we only utilize 19 dances in the first two genres, i.e. Rumba and Cha-cha in this experiment.

TABLE I: Music features for humanoid dances generation

Type	Features	Dimension
Pitch	MFCC, MFCC-delta, constant-Q chromagram	10
Strength	onset strength, tempogram	6
Beat	beat mask, progression within beats	2

2) Music Feature Extraction: As shown in Fig. 2 and Table I, music features m_i are extracted from the audio files, which are composed of pitch features, strength features, and beat features in this work. These features are concatenated as the music feature m_i with 18 dimensions, i.e. $N_m = 18$, and 100 pieces of features are obtained in 1s audio. An audio analysis library named *librosa* [28] is utilized to extract music acoustic features mentioned above.

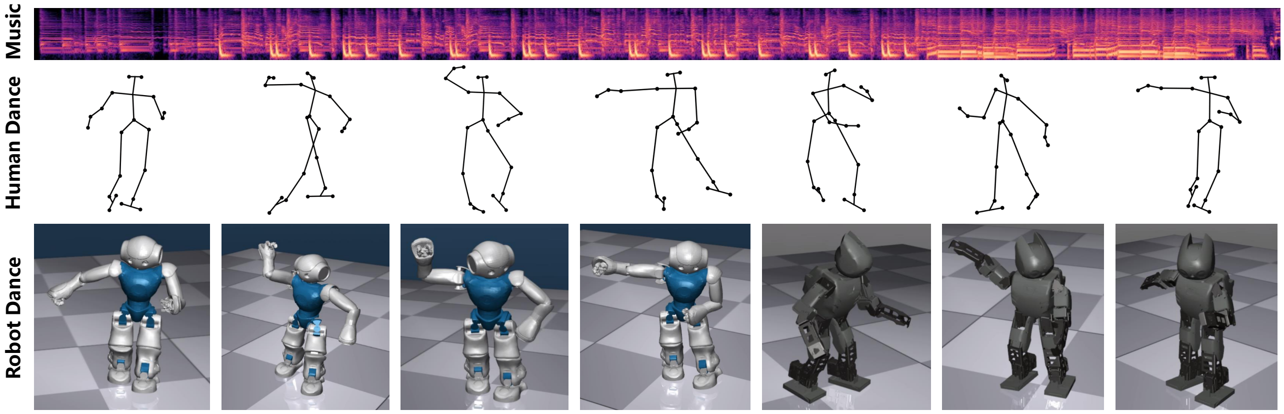

Fig. 3: The dances synthesized for NAO and ROBOTIS OP2 robot in the simulation environment. DanceHAT is trained in the simulation environment for about 2500 epochs. The synthesized dances are similar to human dances and satisfy robot stability requirements after training. More details are given in the video attached.

3) Pose Feature Extraction: In this experiment, the human dance dataset provides skeleton frames of the human, represented by 3D coordinates of 21 key body points. For the convenience of the following training procedure and inspired by recent motion retargeting researches [29], each revolute joint of the robot $\mathcal{R}$ can be easily visually assigned with a corresponding human joint. The robot pose feature $p_i^{\mathcal{R}}$ is defined as angle values of the robot joints, while the human pose feature $p_i^{\mathcal{H}}$ is defined as angle values of the corresponding human joints, which can be calculated by coordinates of key body points. Other retargeting methods and pose feature definitions are also acceptable in this work, such as C-3PO method proposed in [30].

C. Simulation Experiment

In this section, we train and evaluate DanceHAT in the simulation environment. Before the adversarial training procedure, the generator G is pre-trained to imitate human dances directly using similarity loss $\mathcal{L}_{sim}$ only. In the adversarial training procedure, the generator G is trained with Eq. (9). The hyper-parameters α and β are adjusted dynamically to maintain balances between loss gradients. In this work, $\alpha_t = 1.0$, and

$$\beta_t = \omega\left[\mathbb{1}\left(D(G(m_i))_t > \delta\right)\cdot D\left(G(m_i)_t\right) + \varepsilon\right], \qquad (11)$$

where $\mathbb{1}(\cdot)$ denotes unit step function, $\omega = 0.2$, $\delta = 0.4$, and $\varepsilon = 10^{-3}$ in the experiment. During experiments, we find that $\|\frac{\nabla\mathcal{L}_{sim}}{\nabla G}\| < \|\frac{\nabla\mathcal{L}_{stable}}{\nabla G}\|$ generally, because G is pre-trained with $\mathcal{L}_{sim}$, which leads to $\|\frac{\nabla\mathcal{L}_{sim}}{\nabla G}\| \to 0$, especially at the beginning of the training. This leads to $\mathcal{L}_{stable}$ being dominant and the robot preferring to stand still with less movements for stability. Thus, α is designed bigger than β to maintain balance and encourage movements of the robot. Besides, when $D(G(m_i))_t \leq \delta$, i.e. the synthesized dance is predicted stable at time t, β_t is decreased to encourage G to emphasize similarity loss. Otherwise, β_t is increased to prevent the robot from falling down influenced by the similarity loss $\mathcal{L}_{sim}$.

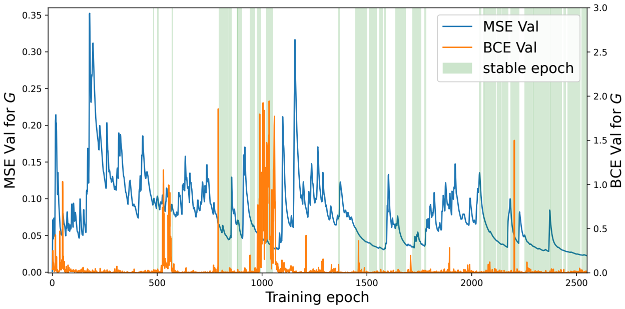

Fig. 4: The MSE loss and BCE loss for G during training on the NAO robot.

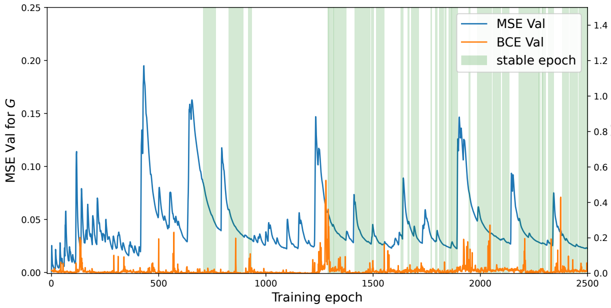

Fig. 5: The MSE loss and BCE loss for G during training on the ROBOTIS OP2 robot.

D. Experiment Result

1) Performance Metrics: In this paper, the following metrics are used to evaluate performance of the models:

- **Completion ratio:** It is defined as the ratio of stable dance length to the whole dance length, i.e. $t_f/T_i \in [0,1]$. It is used to evaluate the stability quality of the generated dances. Higher completion ratio means better stability quality in this work.
- **MSE value:** We use MSE between synthesized dances and corresponding human dances to evaluate per-

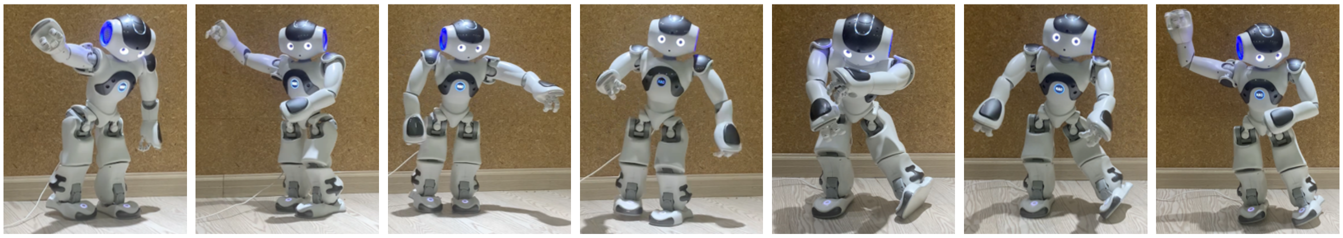

Fig. 6: The experiment result on a real humanoid robot NAO V6. The robot can dance stably with pose sequences generated by DanceHAT. More details of the real robot dances are shown in the video attached.

formance on similarity, i.e. $MSE(p_i^{\mathcal{R}}, p_i^{\mathcal{H}}) \in [0, +\infty]$. Smaller MSE means better performance on similarity in this work. Theoretically, the best result $MSE(p_i^{\mathcal{R}}, p_i^{\mathcal{H}}) = 0$ is impossible, because of the structure differences between the robots and human dancers.

2) Performance During Adversarial Training: The DanceHAT model is trained for about 2500 epochs, i.e. $K = 2500$. Take a piece of music for example, the MSE loss $\mathcal{L}_{sim}$ and BCE loss $\mathcal{L}_{stable}$ for G during training on NAO and ROBOTIS OP2 are described in Fig. 4 and Fig. 5 correspondingly. The epochs in green regions mean that the robot maintains stability during dancing with the synthesized dance. These epochs are called “stable epochs” in the following description for convenience.

Take the training procedure shown in Fig. 5 as an example, the robot cannot maintain stability initially, with the MSE loss closing to 0 because of the pre-training with $\mathcal{L}_{sim}$. After training around 750 epochs, DanceHAT achieves the first stable dance synthesis, with quite a high MSE loss. At this epoch, the generator can synthesize stable dances with small movements. Afterward, DanceHAT tries to synthesize stable dances with lower MSE loss values utilizing adversarial training. As shown in the figure, the model switches between “stable epochs” (green regions) and “unstable epochs” (white regions). In this stage, the generator tries to find a better balance between the similarity and stability with adversarial training, corresponding to $\mathcal{L}_{sim}$ and $\mathcal{L}_{stable}$ respectively. During the “stable epochs”, the MSE loss decreases and BCE loss increases, while in the “unstable epochs” stages, the BCE loss decreases and MSE loss increases. At around 2500th epoch, the synthesized dance is stable with MSE value low enough, which is satisfactory on the similarity and stability metrics.

TABLE II: Comparison of DanceHAT to other methods.

Robot	Metric	Direct	G-only	DanceHAT
NAO	MSE	0.0	9.8×10^{-4}	0.024
	Completion ratio	2.22%	2.36%	100%
ROBOTIS OP2	MSE	0.0	4.1×10^{-4}	0.023
	Completion ratio	0.15%	0.15%	100%

3) Experiment Result: In this section, we compare DanceHAT with other methods to demonstrate effectiveness of our model. As shown in Table II, “**direct**” means transfer human dances to robot dances directly without tuning. “**G-only**” means disabling the discriminator D and training G with $\mathcal{L}_{sim}$ only. The result shows that “direct” and “G-only” have excellent performance on similarity but poor performance on stability with quite small completion ratios. DanceHAT takes a balance between similarity and stability, which obtains excellent completion ratios and satisfactory MSE values.

4) Robot Dance in the Simulation Environment: As shown in Fig. 3, the NAO robot can dance stably in the simulation environment. The dances synthesized are similar to corresponding human dances under the robot stability condition. Because of the constraints of robot movement capability and stability, the synthesized movements of the lower body decrease obviously compared to the human dances, while the upper body movements are same to the human dancers basically. More details are given in the video attached.

E. Experiment on the Real Humanoid Robot

To evaluate performance on the real humanoid robot, experiments on a real NAO robot are conducted. The model trained in the simulation environment is transferred to a NAO V6 robot. As shown in Fig. 6, the real robot can dance stably with the synthesized pose sequences. The real NAO robot can wave arms, turn around slowly, and stand with edges of feet. Some human movements such as whole body rotation and jumping are not in NAO’s capability, thus DanceHAT learns similar robot movements to replaced those action sequence through adversarial training. The experiment results demonstrate that our method is effective for robot dance synthesis and can be applied to dance synthesis tasks with real robots. More details of the experiment results on the real robot are given in the video attached.

V. CONCLUSIONS

In this work, we propose a new method *DanceHAT* for humanoid dance synthesis driven by musics. DanceHAT incorporates similarity and robot stability simultaneously based on adversarial training. Therefore, it can synthesize robot dances and maintain robot stability automatically across different robot platforms. The experiments in the simulation environment and on the real robot demonstrate the effectiveness of our method. Besides, DanceHAT has potential to be utilized in more robot imitation tasks with stability requirements, which will be researched in future works.

References

[1] H. Ahn, J. Kim, K. Kim, and S. Oh, “Generative autoregressive networks for 3d dancing move synthesis from music,” *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3501–3508, 2020.

[2] D. F. P. Granados, B. A. Yamamoto, H. Kamide, J. Kinugawa, and K. Kosuge, “Dance teaching by a robot: Combining cognitive and physical human–robot interaction for supporting the skill learning process,” *IEEE Robotics and Automation Letters*, vol. 2, no. 3, pp. 1452–1459, 2017.

[3] R. Qin, C. Zhou, H. Zhu, M. Shi, F. Chao, and N. Li, “A music-driven dance system of humanoid robots,” *International Journal of Humanoid Robotics*, vol. 15, no. 05, p. 1850023, 2018.

[4] K. Kojima, S. Nozawa, K. Okada, and M. Inaba, “Dance-like humanoid motion generation through foot touch states classification,” in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 1788–1793.

[5] S. F. de Sousa Junior and M. F. M. Campos, “Shall we dance? a music-driven approach for mobile robots choreography,” in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2011, pp. 1974–1979.

[6] G. Xia, J. Tay, R. Dannenberg, and M. Veloso, “Autonomous robot dancing driven by beats and emotions of music,” in *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 1*, 2012, pp. 205–212.

[7] J.-J. Aucouturier, K. Ikeuchi, H. Hirukawa, S. Nakaoka, T. Shiratori, S. Kudoh, F. Kanehiro, T. Ogata, H. Kozima, H. G. Okuno, *et al.*, “Cheek to chip: Dancing robots and ai’s future,” *IEEE Intelligent Systems*, vol. 23, no. 2, pp. 74–84, 2008.

[8] T. Bi, P. Fankhauser, D. Bellicoso, and M. Hutter, “Real-time dance generation to music for a legged robot,” in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 1038–1044.

[9] S. Nakaoka, A. Nakazawa, F. Kanehiro, K. Kaneko, M. Morisawa, and K. Ikeuchi, “Task model of lower body motion for a biped humanoid robot to imitate human dances,” in *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2005, pp. 3157–3162.

[10] T. Tang, J. Jia, and H. Mao, “Dance with melody: An lstm-autoencoder approach to music-oriented dance synthesis,” in *Proceedings of the 26th ACM international conference on Multimedia*, 2018, pp. 1598–1606.

[11] S. Nakaoka, A. Nakazawa, F. Kanehiro, K. Kaneko, M. Morisawa, H. Hirukawa, and K. Ikeuchi, “Learning from observation paradigm: Leg task models for enabling a biped humanoid robot to imitate human dances,” *The International Journal of Robotics Research*, vol. 26, no. 8, pp. 829–844, 2007.

[12] H.-Y. Lee, X. Yang, M.-Y. Liu, T.-C. Wang, Y.-D. Lu, M.-H. Yang, and J. Kautz, “Dancing to music,” in *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, 2019, pp. 3586–3596.

[13] G. Sun, Y. Wong, Z. Cheng, M. S. Kankanhalli, W. Geng, and X. Li, “Deepdance: music-to-dance motion choreography with adversarial learning,” *IEEE Transactions on Multimedia*, vol. 23, pp. 497–509, 2020.

[14] R. Li, S. Yang, D. A. Ross, and A. Kanazawa, “Ai choreographer: Music conditioned 3d dance generation with aist++,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2021, pp. 13 401–13 412.

[15] D. Gouaillier, V. Hugel, P. Blazevic, C. Kilner, J. Monceaux, P. Lafourcade, B. Marnier, J. Serre, and B. Maisonnier, “Mechatronic design of nao humanoid,” in *2009 IEEE International Conference on Robotics and Automation*. IEEE, 2009, pp. 769–774.

[16] I. Ha, Y. Tamura, H. Asama, J. Han, and D. W. Hong, “Development of open humanoid platform darwin-op,” in *SICE Annual Conference 2011*. IEEE, 2011, pp. 2178–2181.

[17] C. Guo, X. Zuo, S. Wang, S. Zou, Q. Sun, A. Deng, M. Gong, and L. Cheng, “Action2motion: Conditioned generation of 3d human motions,” in *Proceedings of the 28th ACM International Conference on Multimedia*, 2020, pp. 2021–2029.

[18] X. Ren, H. Li, Z. Huang, and Q. Chen, “Self-supervised dance video synthesis conditioned on music,” in *Proceedings of the 28th ACM International Conference on Multimedia*, 2020, pp. 46–54.

[19] R. Huang, H. Hu, W. Wu, K. Sawada, M. Zhang, and D. Jiang, “Dance revolution: Long-term dance generation with music via curriculum learning,” in *International Conference on Learning Representations*, 2021. [Online]. Available: https://openreview.net/forum?id=xGZG2kS5bFk

[20] M. Vukobratovic, B. Borovac, D. Surla, and D. Stokic, *Biped locomotion: dynamics, stability, control and application*. Springer Science & Business Media, 2012, vol. 7.

[21] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” *Advances in neural information processing systems*, vol. 27, 2014.

[22] A. Radford, L. Metz, and S. Chintala, “Unsupervised representation learning with deep convolutional generative adversarial networks,” *arXiv preprint arXiv:1511.06434*, 2015.

[23] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, “Image-to-image translation with conditional adversarial networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1125–1134.

[24] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, “Domain-adversarial training of neural networks,” *The journal of machine learning research*, vol. 17, no. 1, pp. 2096–2030, 2016.

[25] C. Yu and A. Tapus, “Srg 3: Speech-driven robot gesture generation with gan,” in *2020 16th International Conference on Control, Automation, Robotics and Vision (ICARCV)*. IEEE, 2020, pp. 759–766.

[26] K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, “Learning phrase representations using RNN encoder–decoder for statistical machine translation,” in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 1724–1734. [Online]. Available: https://aclanthology.org/D14-1179

[27] E. Todorov, T. Erez, and Y. Tassa, “Mujoco: A physics engine for model-based control,” in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012, pp. 5026–5033.

[28] B. McFee, C. Raffel, D. Liang, D. P. Ellis, M. McVicar, E. Battenberg, and O. Nieto, “librosa: Audio and music signal analysis in python,” in *Proceedings of the 14th python in science conference*, vol. 8. Citeseer, 2015, pp. 18–25.

[29] L. Penco, B. Clément, V. Modugno, E. M. Hoffman, G. Nava, D. Pucci, N. G. Tsagarakis, J.-B. Mouret, and S. Ivaldi, “Robust real-time whole-body motion retargeting from human to humanoid,” in *2018 IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids)*. IEEE, 2018, pp. 425–432.

[30] T. Kim and J.-H. Lee, “C-3po: Cyclic-three-phase optimization for human-robot motion retargeting based on reinforcement learning,” in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 8425–8432.