

# DFM: Deep Fourier Mimic for Expressive Dance Motion Learning

Ryo Watanabe<sup>1,2</sup>, Chenhao Li<sup>1,3</sup> and Marco Hutter<sup>1</sup>

**Abstract**—As entertainment robots gain popularity, the demand for natural and expressive motion, particularly in dancing, continues to rise. Traditionally, dancing motions have been manually designed by artists, a process that is both labor-intensive and restricted to simple motion playback, lacking the flexibility to incorporate additional tasks such as locomotion or gaze control during dancing. To overcome these challenges, we introduce Deep Fourier Mimic (DFM), a novel method that combines advanced motion representation with Reinforcement Learning (RL) to enable smooth transitions between motions while concurrently managing auxiliary tasks during dance sequences. While previous frequency domain based motion representations have successfully encoded dance motions into latent parameters, they often impose overly rigid periodic assumptions at the local level, resulting in reduced tracking accuracy and motion expressiveness, which is a critical aspect for entertainment robots. By relaxing these locally periodic constraints, our approach not only enhances tracking precision but also facilitates smooth transitions between different motions. Furthermore, the learned RL policy that supports simultaneous base activities, such as locomotion and gaze control, allows entertainment robots to engage more dynamically and interactively with users rather than merely replaying static, pre-designed dance routines.

## I. INTRODUCTION

Recent advancements in entertainment robots [1]–[3] have significantly broadened their capabilities, enabling them to perform a wider range of tasks and facilitating more meaningful human-robot interactions [4]–[6]. A prominent example is Sony’s aibo [1], a robotic dog designed to engage and captivate its owners. Within the realm of entertainment robotics, dancing has become one of the most effective ways to attract human attention and express robotic emotions through motion [7]–[9].

Typically, the diverse artistic motions performed by these robots are meticulously crafted by designers at entertainment robotics companies [3], [10]. This process is inherently time-consuming, requiring motion designers to carefully create and fine-tune various dance motions, most of which are handcrafted. Despite the presence of common motion primitives within these dance routines, designers still need to create complete dance motions for each instance, which demands significant time and effort. Additionally, these artistic motions are often replayed with less emphasis on interacting dynamically with humans.

The field of computer animation has proposed various methods to streamline the motion creation process. A promising approach involves the use of learning-based methods to

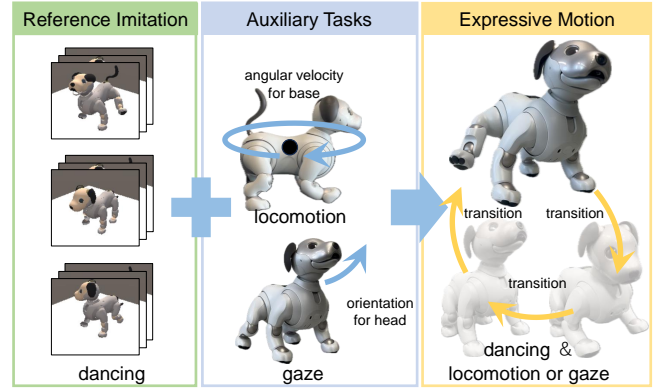


Fig. 1. Deep Fourier Mimic (DFM) allows entertainment robots such as aibo to seamlessly combine artistic motions crafted by designers with auxiliary tasks like locomotion or gaze towards a human face, resulting in expressive motion that can smoothly transition between different movements at arbitrary timings. Project webpage: <https://sony.github.io/DFM/>

generate diverse motions. Recent advancements in this area have leveraged Generative Adversarial Networks (GANs) [11] to address this issue [12]–[18]. However, these methods still face limitations in handling a wide variety of demonstration motion datasets due to mode collapse. An alternative approach focuses on motion representation. Several studies [19]–[22] have explored methods to automatically align and represent periodic dance motions within a structured latent space. For instance, the Fourier Latent Dynamics (FLD) approach [22], which integrates Periodic Autoencoder (PAE) [19] and RL, addresses some of these challenges by conditioning desired motion trajectories during policy inference. The structured latent space in FLD allows for the creation of diverse motions through smooth transitions between motion primitives, thereby reducing the time-intensive process of motion creation for designers. However, the reliance on strong periodicity assumptions in FLD, even at the local level, leads to overly smoothed motions and a loss of expressiveness, which is particularly problematic for entertainment robots where dynamic and expressive motions are essential.

To extend the capabilities of FLD to general, less periodic motions such as dancing, we propose DFM as shown in Fig. 1, a method that trains policies using fresh encodings of the most recent motion segments, thereby relaxing the strong local periodic assumptions to preserve expressiveness and variation in the generated motions. Additionally, to increase human interaction during stylized dancing, we introduce supplementary tasks such as locomotion and head orientation tracking called gaze. For instance, maintaining eye contact by gazing at a human face, which mimics real dog behavior, can create a more interactive and engaging experience during

<sup>1</sup>Ryo Watanabe, Chenhao Li and Marco Hutter are with the Robotic Systems Lab, Department of Mechanical Engineering, ETH Zurich, Switzerland.

<sup>2</sup>Ryo Watanabe is with Sony Group Corporation, Japan

<sup>3</sup>Chenhao Li is with ETH AI Center, Switzerland

the dance with a whole-body approach.

Our primary contributions are as follows:

- Development of a method for learning from demonstrations that achieves high tracking accuracy of reference motions while preserving details and ensuring smooth transitions between various dance sequences for entertainment robots.
- Demonstration and validation in hardware experiments of multi-task capabilities, including auxiliary gaze and locomotion control during dance motion, coordinated in a whole-body manner that integrates both leg and head movements.

## II. RELATED WORK

### A. Robotic Dance

The development of dancing for entertainment robots has seen significant progress [7]. Nakaoka et al. employed a motion capture system to teach the HRP-2 [23] a traditional Japanese folk dance [24]. While this approach showcased impressive dancing capabilities, the sequences and timings of all movements were meticulously pre-programmed to align with a specific music. For quadruped robots, a notable example is a dancing robot capable of detecting tempo and adjusting its motions in real-time [9]. Similarly, certain entertainment robots, such as Sony’s aibo, incorporate the ability to adjust their dancing tempo [10]. However, these systems rely mostly on replaying predefined motion sequences and lack the capability to perform additional task operations during motion execution, which is a key focus of this paper.

### B. Learning from Demonstrations

Reinforcement learning (RL) has been extensively utilized to generate robust locomotion policies for quadrupedal robots [25]–[27]. Despite these advances, achieving stylized motions that appear natural to human observers remains challenging. To address the need for natural motion learning, Peng et al. [28] proposed a method that enhances RL by incorporating rewards to imitate reference trajectories within a physics-based simulator. Recently, Ruben et al. [3] applied this approach to real robots. However, these methods are constrained by their reliance on a single trajectory, making it difficult to transition between reference motions without intricate motion design.

To handle transitions between multiple reference motions, various methods based on GANs have been developed [12], [13]. Nevertheless, these approaches are often plagued by mode collapse, a phenomenon resulting from insufficient motion distinction without intrinsic diversity rewards. Although recent works [15]–[18] have made efforts to mitigate mode collapse, they still struggle to handle the switching of specific motions at arbitrary timings, a critical requirement for entertainment robots.

In contrast, self-supervised representation methods using physics simulators have emerged as another promising research direction [29]. While some previous studies have specifically focused on learning dancing motions [20], [21], the PAE [19] has proven effective in producing smoother

and more stable movements for periodic motions, such as locomotion and dancing, with a special treatment of the spatial-temporal relationships inherent in the motions. The FLD approach [22] further extends PAE by introducing latent dynamics with a quasi-constant representation of motions that greatly reduces the parameterization effort in motion representation and facilitates downstream learning with RL. These methods automatically align and represent motions within a structured latent space, allowing for conditioning desired motion trajectories during policy inference without suffering from mode collapse. However, both the representation and policy learning in FLD impose strong periodic assumptions, which can lead to over-smoothed motions that lack expressiveness, a significant drawback for entertainment robots that aim to replicate the nuanced motions of real animals.

## III. METHOD

As illustrated in Fig. 2, DFM training pipeline for expressive dance motion learning system comprises four key components: motion design, motion representation, RL training, and hardware inference.

### A. Motion Design

Reference motions, including joint positions and velocities for each joint, are created using specialized design tools. Following practices in the animation industry, our motion data are crafted by artistic designers. We select 34 distinct dance motions, all created by motion designers at Sony, as showcased in the supplementary video. Each motion is augmented to include five frequency variations — 0.5, 0.75, 1.0, 1.25, and 1.5 — by sampling and interpolating the motion trajectories. This process results in a total of 170 clips with a duration of 6 seconds each.

### B. Motion Representation

Motions are commonly described as long-horizon trajectories in high-dimensional state space. However, directly associating motions with raw trajectory instances yields highly inefficient representations and poor generalization, failing to develop a policy that naturally transits in between. We denote trajectory segments of length  $H$  in  $d$ -dimensional state space preceding time step  $t$  by  $\mathbf{s}_t = (s_{t-H+1}, \dots, s_t) \in \mathbb{R}^{d \times H}$ . In the motion representation stage, we utilize FLD [22] to encode motion trajectories into latent parameters, which consist of  $\theta_t = (f_t, a_t, b_t)$  and  $\phi_t$ , where  $f_t$ ,  $a_t$ ,  $b_t$  and  $\phi_t$  denote latent frequency, amplitude, offset and phase, respectively. The PAE structure featured by FLD employs a set of encoder and decoder composed of 1D convolutional layers through time [19]. While  $\theta_t$  are computed with a differentiable real Fast Fourier Transform (FFT) layer, the latent phase  $\phi_t$  is determined using a linear layer followed by atan2 applied on 2D phase shifts on each channel as shown in Fig. 2. With the characteristic convolution and FFT layers, the FLD encoder decomposes the input motions into a latent phase parameter  $\phi_t$  representing the low-level local time index and a latent frequency domain parameter  $\theta_t$  representing high-level global

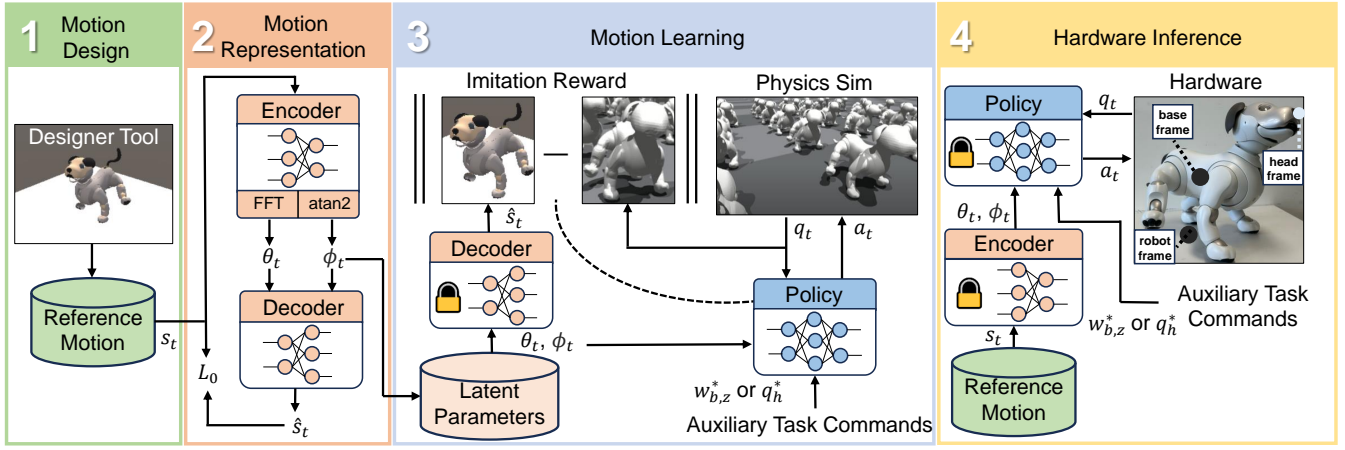


Fig. 2. The expressive dance motion learning system is composed of four key components: motion design, motion representation, motion learning, and hardware inference. In the motion design phase, artists create motion references using specialized design software. The representation of these diverse motions is then learned using a PAE. Reinforcement learning (RL) is employed to enable the robot to perform auxiliary tasks, such as walking and head orientation control, while accurately tracking the designed dance references. During inference, the learned policy is deployed on the actual hardware, allowing for real-time execution of dance motions and dynamic and interactive motions by tracking the auxiliary task commands.

features, respectively. In the original paper, FLD formulates latent dynamics with latent frequency  $f_t$  and time increment  $\Delta t$  is thus described as

$$\theta_t = \theta_{t-1}, \quad \phi_t = \phi_{t-1} + f_{t-1} \Delta t. \quad (1)$$

The encoder and decoder are described with  $c$  latent channels,

$$\phi_t, \theta_t = \mathbf{enc}(s_t), \quad \hat{s}'_{t+i} = \mathbf{dec}(\phi_t + i f_t \Delta t, \theta_t), \quad (2)$$

where  $\phi_t \in \mathbb{R}^c$  and  $\theta_t \in \mathbb{R}^{3c}$ .

Fourier Latent Dynamics (FLD) introduces a forward prediction mechanism in the latent representation space, where it reconstructs future states with phase propagation while maintaining  $\theta_t$ . The proceeding motion segment  $s_{t+i} = (s_{t-H+1+i}, \dots, s_{t+i})$  is approximated with the prediction  $\hat{s}'_{t+i}$  decoded from  $i$ -step forward propagation using the latent dynamics from time step  $t$ . The latent dynamics in Eq. 1 assumes locally constant latent parameterizations and propagates latent states by advancing  $i$  local phase increments. Therefore, the total loss for training FLD with  $N$ -step forward prediction by propagating the latent dynamics is written as

$$L_{FLD}^N = \sum_{i=0}^N \text{MSE}(\hat{s}'_{t+i}, s_{t+i}). \quad (3)$$

The choice of the number of forward prediction steps,  $N$ , plays a crucial role in balancing the trade-off between the accuracy of local reconstructions and the global coherence of the latent parameterizations. In the original FLD approach, a larger value of  $N$ , such as  $N = 100$  at  $\Delta t = 0.01$ , is typically selected to emphasize the global consistency of the latent parameters within periodic motions to facilitate downstream RL training. However, this can result in reduced reconstruction accuracy of general, less periodic reference trajectories, which is particularly critical for entertainment robots where precise motion tracking is essential. To address this, DFM sets  $N = 0$ , prioritizing local reconstruction accuracy to enhance the tracking performance while maintaining a fresh local motion

TABLE I  
MOTION REPRESENTATION HYPER PARAMETERS

Entry	Symbol	Value
step time seconds	$\Delta t$	0.01
mini batch size	—	50
learning rate	—	0.0001
weight decay	—	0.0005
max iterations	—	5000
latent channel number	$c$	8
periodic trajectory segment	$H$	100
forward prediction step	$N$	0

TABLE II  
POLICY OBSERVATION

Entry	Symbol	Noise Level	Dimensions
joint positions	$q$	0.01	14
joint velocities	$\dot{q}$	1.5	14
last action	$a^*$	0.0	14
foot contact state	$f_c$	0.0	4
gravity orientation	$g$	0.05	3
latent phase	$\sin \phi$	0.0	8
latent phase	$\cos \phi$	0.0	8
latent frequency	$f$	0.0	8
latent amplitude	$a$	0.0	8
latent offset	$b$	0.0	8

encoding during policy training. The specific hyperparameters used for the motion representation are detailed in Table I.

### C. Motion Learning

Following the motion representation stage, DFM employs RL to track the encoded motions while managing additional tasks such as locomotion and gaze control during stylized dancing, as depicted in Fig. 2. The training network architecture consists of three fully connected layers, each with 256 hidden units and ELU activation functions, and utilizes the

TABLE III  
REWARD FUNCTIONS AND SCALES AT IMITATION AND CURRICULUM PHASE FOR LOCOMOTION AND GAZE

Category	Reward	Definition	Dance imitation	Locomotion curriculum	Gaze curriculum
<b>Imitation</b>	joint position imitation	$\exp(\ q^* - q\ ^2)$	1.0	1.0	1.0
<b>Task</b>	base angular velocity tracking	$\exp(-\frac{1}{0.06}\ w_{b,z}^* - w_{b,z}\ ^2)$	0.0	1.0	-
	end-effector orientation tracking	$\exp(4\ q_h^* - q_h\ ^2)$	0.0	-	0.7
<b>Regularization</b>	joint torque	$\ \tau\ ^2$	-0.001	-0.001	-0.001
	joint acceleration	$\ \ddot{q}\ ^2$	$-2e^{-7}$	$-2e^{-7}$	$-2e^{-7}$
	joint target difference	$\ a_{t-1}^* - a_t^*\ ^2$	-0.01	-0.01	-0.01
	self-collisions	$n_c$	-10.0	-10.0	-10.0
	foot slippage	$\ v_{f,xy}\ ^2$	0.0	-0.15	-
	foot air time	$\sum_i (t_{f,air} - 0.2)$	0.0	2.0	-

Proximal Policy Optimization (PPO) [30]. Both the actor and critic networks are implemented as Multi-Layer Perceptrons (MLPs), adhering to the architecture and hyperparameters established in prior work [22]. The simulation and control loop frequencies are set to 400 Hz and 100 Hz, respectively, using the Isaac Gym framework [31].

The aibo features 14 Degrees of Freedom (DoF), including 12 DoF for its legs and 2 DoF for head pitch and yaw movements, represented by the action vector  $a^*$ . Policy observations and noise levels are detailed in Table II. Given that aibo is a consumer-grade robot,  $f_c$  values are obtained from binary switch contact sensors. The latent parameters with 8 channels are used as observations. Unlike FLD, which assumes time-invariant latent parameters  $\theta_t$  for each episode, DFM updates  $\theta_t$  and  $\phi_t$  dynamically during motion representation to capture the nuanced details of the artistic reference motions. This allows for the expression of critical changes in frequency, amplitude, and offset, preserving details in expressive movements.

In contrast to FLD, which limits motion representation and learning only to locomotion tasks, the extended capabilities of DFM are demonstrated with distinct tasks assigned to each stage. Specifically during the RL phase, the policy is conditioned to track a target motion sequence using the learned representations, while locomotion and gaze are trained with auxiliary task objectives. The task command for locomotion is the angular velocity of the base axis, while for gaze is the pitch and yaw angles of the head axis in the robot frame, as illustrated in Fig. 2.

The reward functions and their corresponding scales are outlined in Table III. The variables  $q$ ,  $\tau$ , and  $n_c$  represent joint positions, joint torques, and the number of collisions, respectively.  $q_h$  denotes the yaw and pitch angles of the head axis, with  $q_h^*$  representing their commanded values. Similarly,  $w_b$  and  $w_b^*$  denote the measured and desired angular velocities of the base axis, respectively. Imitation rewards are designed to ensure the policy closely mimics the reference dancing motion. Joint imitation rewards are computed as the difference between the reconstructed joint positions and the current joint position in the physics simulator. Once the agent achieves a joint position imitation reward higher than 0.9,

task rewards for locomotion or gaze are introduced. Among the regularization rewards, joint target difference, joint torque, and joint acceleration rewards suppress jerky motions, while self-collision rewards prevent collisions between the robot's parts. Foot velocities  $v_f$  for foot slippage and airtime for foot parts  $t_{f,air}$  are monitored to shape walking during the locomotion curriculum phase.

#### D. Hardware Inference

As depicted in Fig. 2, the reference motion, encoder, and policy, developed during the motion design, motion representation, and motion learning stages, respectively, are deployed on the robot. The RL policies for locomotion and gaze are implemented as separate entities, with task commands being switched accordingly for each policy. Both control and sensor reading operations are performed at a frequency of 100 Hz. The gravity orientation is calculated by fusing acceleration and gyroscope data from the IMU.

To ensure smooth transitions between different motions at arbitrary times, we interpolate the latent parameters during the transition period using Eq. 4:

$$\begin{aligned}\theta_{A \rightarrow B} &= \alpha \theta_A + (1 - \alpha) \theta_B, \\ \phi_{A \rightarrow B} &= \alpha \phi_A + (1 - \alpha) \phi_B,\end{aligned}\tag{4}$$

where  $\theta_A$  and  $\phi_A$  represent the final latent parameters of motion A, and  $\theta_B$  and  $\phi_B$  correspond to the initial latent parameters of motion B. The interpolation factor  $\alpha$  is smoothly varied from 0 to 1 over a designated transition period, which in our implementation is set to 0.5 seconds.

## IV. EXPERIMENTS AND RESULTS

We evaluate the performance of DFM in terms of tracking accuracy relative to reference motions and the naturalness of transitions between motions. Additionally, we demonstrate the multitasking capabilities of our approach through locomotion and gaze control during stylized dancing.

#### A. Tracking Accuracy

To assess tracking accuracy, we calculate the difference between the reference motion and the observed joint positions on the robot hardware. For this evaluation, we select a dancing motion that involved lifting the rear legs. Fig. 3 illustrates

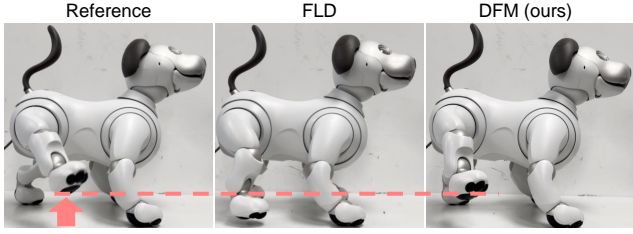


Fig. 3. Height reached by rear right leg. Left, middle and right depict reference motion, FLD and DFM motions respectively. The red dash line illustrates the height of the right rear leg at reference motion.

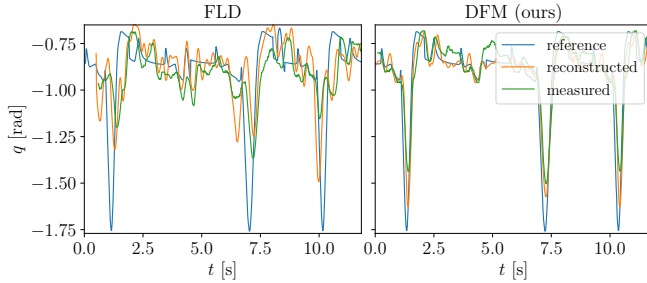


Fig. 4. Comparison of tracking accuracy for the FLD and DFM. Blue: reference motion created by the motion designer. Orange: reconstructed motions from motion representation parts by conditioning the reference motion. Green: joint encoder reading activated by the RL policy.

the height of the rear right leg during this stylized dance. Although DFM does not perfectly replicate the height of the reference motion, it significantly outperforms the FLD baseline. To provide a more quantitative comparison between the baseline and our method, we analyze three types of joint angles as shown in Fig. 4. The results indicate that the motion reconstructed by FLD is overly smooth due to its strong enforcement of quasi-constant parameterization and periodicity assumption with  $N = 100$ . In contrast, DFM achieves a more accurate reconstruction with  $N = 0$ , preserving intricate details that may not follow periodic patterns. When examining the joint encoder data measured from the robot, FLD again shows excessive smoothing, which we attribute to the overly strong periodic assumptions applied to the local time during RL training, as described in Eq. 2. Figure 5 presents the  $\sin \phi$  and frequency values derived from the latent parameters across eight channels during the same dancing motion. On the left side, FLD shows that all channels of  $\sin \phi$  are periodic, with little change in frequency. In contrast, DFM demonstrates variability in some channels of  $\sin \phi$  and frequency during the upward movement of the rear leg, retaining non-periodic features that characterize the dance motions. Finally, Table IV reports the mean absolute tracking error (MAE) across all joints for all 170 evaluated motions using the real aibo hardware. Additionally, we test DFM using the MIT Humanoid environment [32] in Isaac Gym. Our method consistently demonstrates superior tracking accuracy in both robot environments compared to FLD.

### B. Natural Transition

The motion representation employed by DFM enables continuous frequency interpolation and smooth transitions

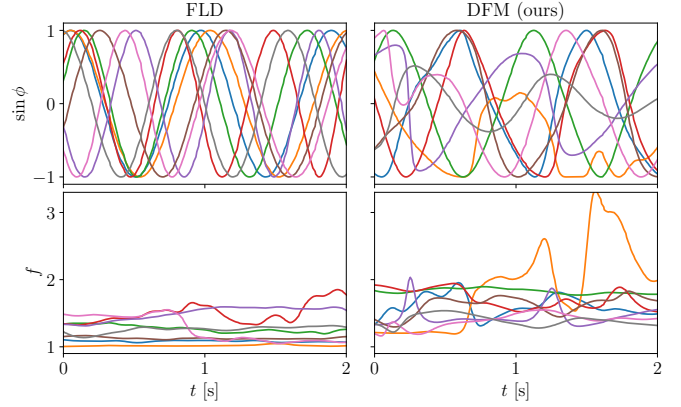


Fig. 5. Comparison of 8 channel latent parameters for FLD at the left and DFM at the right side by conditioning the same dancing motion as Fig. 3. The upper and bottom of plots are  $\sin \phi$  and frequency for each.

TABLE IV  
MEAN ABSOLUTE TRACKING ACCURACY

Robot	reference motion	FLD	DFM (ours)
aibo	dance	0.132 rad	0.094 rad
MIT humanoid	locomotion	0.125 rad	0.103 rad

between different dancing motions.

Figure 6 shows the estimated latent frequency parameters conditioned on the reference motion, which primarily involves head movements transitioning from higher to lower dancing frequencies. While most frequency channels remain relatively constant, channels 3 and 4 exhibit gradual changes as shown in Fig. 6. The linear interpolation of frequencies in these channels adjusts in response to the changing frequency of the reference dancing motion. Even though the training dataset consists of discrete frequency types, the motion representation allows for continuous frequency interpolation. This capability results in smooth, periodic changes in joint positions without abrupt movements, even for previously unseen datasets as shown at the bottom plot in Fig. 6.

Figure 7 illustrates the joint angular velocity at the head pitch and yaw during the transition from motion A to motion B, which primarily involves the head pitch and yaw actuators, as shown in the supplementary video. During the transition times at 1 and 2.5 seconds, joint positions experience abrupt changes with switches between reference motions. We compare the transition performance of DFM with DeepMimic [28], a well-known learning from demonstration approach that yields high tracking performance on single trajectories but lacks capabilities to deal with multiple motions. Jerky transitions are observed in this case if the reference dataset and its representation are not carefully crafted. In contrast, DFM achieves smooth transitions without abrupt movements by interpolating in the latent space using Eq. 4.

### C. Multi-task Demonstration

We evaluate the multitasking capability of DFM with auxiliary tasks, including locomotion and gaze, respectively.



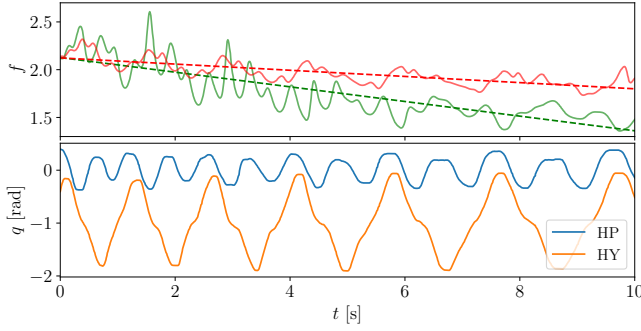


Fig. 6. Frequency modulation during head-moving dance. The upper plot displays the frequency of two representative latent channels out of eight. Solid and dashed curves represent raw and linearly interpolated data, respectively. The bottom plot shows head pitch (HP) and the head yaw (HY) joint angles.

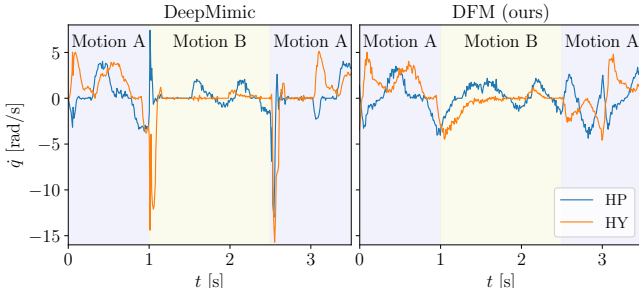


Fig. 7. Transition between different dance types. The background color indicates the dance motion type. The left and right plots demonstrate hard switches between Dance A and Dance B, with DeepMimic and DFM, respectively. Angular velocities at the head pitch (HP) and the head yaw (HY) are shown.

Figure 8 illustrates the locomotion policy during dancing, where an angular velocity command is used to facilitate in-place rotation. In the reference motion, only the rear legs move alternately, while the forelegs remain stationary. After training the policy with the reward structure defined in the locomotion curriculum (Table III), aibo learns to rotate in response to the angular velocity command in the base frame. To allow this rotation without hindering the movement of the rear legs, the right foreleg is lifted, enabling the execution of the stylized dancing, as shown in the supplementary video.

Similarly, a policy for auxiliary gaze control is trained using the reward scale from the gaze curriculum in Table III. This policy enables aibo to adjust its head orientation in response to pitch and yaw commands during dancing, as demonstrated in Fig. 9. The supplementary video shows that the dance sequence continues smoothly while the robot adjusts its pitch and yaw angles. aibo utilizes its head and legs to track the commanded pitch and yaw angles, as illustrated in Fig. 10. For instance, when a pitch of 0.3 rad and a yaw of 0.0 rad are commanded, both the directions of head are moved up with legs. In contrast, a  $-0.5$  rad pitch command prompts the head and legs to move in opposite directions.

## V. CONCLUSION

We introduce DFM, a novel approach that combines motion representation with RL to achieve high tracking accuracy, smooth transitions, and the capability to perform additional tasks during dancing. By relaxing the strong periodic assump-

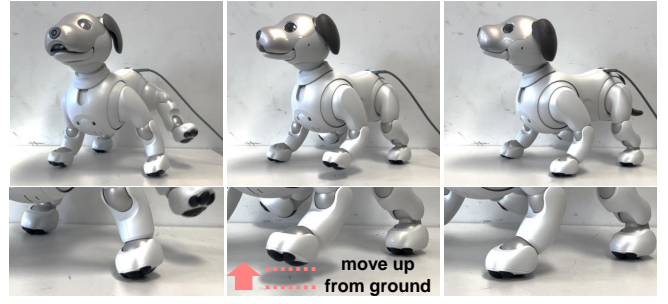


Fig. 8. Locomotion during dance. The reference dance motion alternates lifting the rear legs while keeping the forelegs stationary. Applying an angular velocity command results in locomotion by lifting the left foreleg.



Fig. 9. Gaze during dance. The reference motion is the same as in Fig. 8. The images depict commanded pitch angles of 0.0, 0.3 rad, and  $-0.5$  rad, respectively. The command of the yaw angle is held at zero.

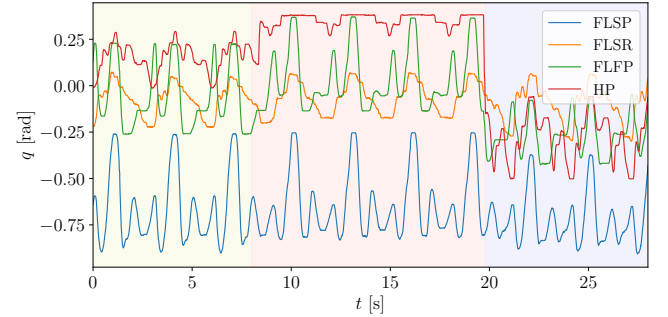


Fig. 10. Joint readings for fore left shoulder pitch (FLSP), fore left shoulder roll (FLSR), fore left foot pitch (FLFP), and head pitch (HP) during dancing gaze are shown. The background color of the plot indicates command for pitch angle at the head frame (yellow: 0.0, red: 0.3 rad, blue:  $-0.5$  rad).

tions presented in previous works, our method demonstrates superior motion expressiveness compared to the baselines. Our motion representation technique enables continuous frequency changes for unseen reference datasets and facilitates smooth transitions between different motion types through latent space interpolation. Additionally, DFM demonstrates extended multi-task capabilities, such as locomotion and gaze control during dancing, leading to more interactive motions beyond simple motion replay. These findings have significant implications for advancing research in human-robot interaction for entertainment robots and open new avenues to create more expressive and interactive robots.

## ACKNOWLEDGMENT

The authors would like to thank Hiroyuki Izumi and Ai Tateishi at Sony Group Corporation for software integration help. Thanks to Takahiro Miki, Yuntao Ma, Kento Kawaharazuka at ETH Zurich for reinforcement learning discussion.

## REFERENCES

- [1] Sony, “aibo,” <https://us.aibo.com/>, online; accessed 7-Feb-2024.
- [2] Grove X, “Lovot,” <https://lovot.life/en/>, online; accessed 7-Feb-2024.
- [3] G. Ruben, K. Espen, A. Michael, W. Georg, B. Jared, P. Steven, M. David, and B. Moritz, “Design and control of a bipedal robotic character,” *Robotics: Science and Systems (RSS)*, 2024.
- [4] H. S. Kang, K. Makimoto, R. Konno, and I. S. Koh, “Review of outcome measures in paro robot intervention studies for dementia care,” *Geriatric Nursing*, vol. 41, no. 3, pp. 207–214, 2020.
- [5] B. Friedman, P. H. Kahn, and J. Hagman, “Hardware companions? what online aibo discussion forums reveal about the human-robotic relationship,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI ’03. New York, NY, USA: Association for Computing Machinery, 2003, p. 273–280.
- [6] H. Robinson, B. MacDonald, N. Kerse, and E. Broadbent, “The psychosocial effects of a companion robot: a randomized controlled trial,” *Journal of the American Medical Directors Association*, vol. 14, no. 9, pp. 661–667, 2013.
- [7] J.-J. Aucouturier, K. Ikeuchi, H. Hirukawa, S. Nakaoka, T. Shiratori, S. Kudoh, F. Kanehiro, T. Ogata, H. Kozima, H. G. Okuno, M. P. Michalowski, Y. Ogai, T. Ikegami, K. Kosuge, T. Takeda, and Y. Hirata, “Cheek to chip: Dancing robots and ai’s future,” *IEEE Intelligent Systems*, vol. 23, no. 2, pp. 74–84, 2008.
- [8] E. Ackerman, “How boston dynamics taught its robots to dance,” <https://spectrum.ieee.org/how-boston-dynamics-taught-its-robots-to-dance>, online; accessed 17-Jul-2024.
- [9] T. Bi, P. Fankhauser, D. Bellicoso, and M. Hutter, “Real-time dance generation to music for a legged robot,” in *IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*, 2018, pp. 1038–1044.
- [10] Sony, “Dance to the rhythm in aibo,” <https://us.aibo.com/feature/rhythmdance.html>, online; accessed 13-Jul-2024.
- [11] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial networks,” *Communications of the ACM*, vol. 63, no. 11, pp. 139–144, 2020.
- [12] X. B. Peng, Z. Ma, P. Abbeel, S. Levine, and A. Kanazawa, “Amp: Adversarial motion priors for stylized physics-based character control,” *ACM Transactions on Graphics (ToG)*, vol. 40, no. 4, pp. 1–20, 2021.
- [13] A. Escontrela, X. B. Peng, W. Yu, T. Zhang, A. Iscen, K. Goldberg, and P. Abbeel, “Adversarial motion priors make good substitutes for complex reward functions. 2022 ieee,” in *IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*, vol. 2, 2022.
- [14] C. Li, M. Vlastelica, S. Blaes, J. Frey, F. Grimmering, and G. Martius, “Learning agile skills via adversarial imitation of rough partial demonstrations,” in *Conference on Robot Learning*. PMLR, 2023, pp. 342–352.
- [15] X. B. Peng, Y. Guo, L. Halper, S. Levine, and S. Fidler, “Ase: Large-scale reusable adversarial skill embeddings for physically simulated characters,” *ACM Transactions On Graphics (TOG)*, vol. 41, no. 4, pp. 1–17, 2022.
- [16] C. Li, S. Blaes, P. Kolev, M. Vlastelica, J. Frey, and G. Martius, “Versatile skill control via self-supervised adversarial imitation of unlabeled mixed motions,” in *2023 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2023, pp. 2944–2950.
- [17] Z. Luo, J. Cao, K. Kitani, W. Xu, *et al.*, “Perpetual humanoid control for real-time simulated avatars,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 10 895–10 904.
- [18] C. Tessler, Y. Kasten, Y. Guo, S. Mannor, G. Chechik, and X. B. Peng, “Calm: Conditional adversarial latent models for directable virtual characters,” in *ACM SIGGRAPH 2023 Conference Proceedings*, 2023, pp. 1–9.
- [19] S. Starke, I. Mason, and T. Komura, “Deepphase: periodic autoencoders for learning motion phase manifolds,” *ACM Trans. Graph.*, vol. 41, no. 4, jul 2022. [Online]. Available: <https://doi.org/10.1145/3528223.3530178>
- [20] R. Li, S. Yang, D. A. Ross, and A. Kanazawa, “Ai choreographer: Music conditioned 3d dance generation with aist++,” in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 13 381–13 392.
- [21] G. Valle-Pérez, G. E. Henter, J. Beskow, A. Holzapfel, P.-Y. Oudeyer, and S. Alexanderson, “Transflower: probabilistic autoregressive dance generation with multimodal attention,” *ACM Trans. Graph.*, vol. 40, no. 6, dec 2021. [Online]. Available: <https://doi.org/10.1145/3478513.3480570>
- [22] C. Li, E. Stanger-Jones, S. Heim, and S. Kim, “Fld: Fourier latent dynamics for structured motion representation and learning,” *arXiv preprint arXiv:2402.13820*, 2024.
- [23] K. Kaneko, F. Kanehiro, S. Kajita, H. Hirukawa, T. Kawasaki, M. Hirata, K. Akachi, and T. Isozumi, “Humanoid robot hrp-2,” in *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA ’04. 2004*, vol. 2, 2004, pp. 1083–1090 Vol.2.
- [24] 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*, 2005, pp. 3157–3162.
- [25] J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, “Learning quadrupedal locomotion over challenging terrain,” *Science Robotics*, vol. 5, no. 47, p. eabc5986, 2020.
- [26] T. Miki, J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, “Learning robust perceptive locomotion for quadrupedal robots in the wild,” *Science Robotics*, vol. 7, no. 62, p. eabk2822, 2022.
- [27] S. Choi, G. Ji, J. Park, H. Kim, J. Mun, J. H. Lee, and J. Hwangbo, “Learning quadrupedal locomotion on deformable terrain,” *Sci Robot*, vol. 8, no. 74, p. eade2256, Jan. 2023.
- [28] X. B. Peng, P. Abbeel, S. Levine, and M. Van de Panne, “Deepmimic: Example-guided deep reinforcement learning of physics-based character skills,” *ACM Transactions On Graphics (TOG)*, vol. 37, no. 4, pp. 1–14, 2018.
- [29] D. Holden, J. Saito, and T. Komura, *A Deep Learning Framework for Character Motion Synthesis and Editing*, 1st ed. New York, NY, USA: Association for Computing Machinery, 2023. [Online]. Available: <https://doi.org/10.1145/3596711.3596789>
- [30] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, “Proximal policy optimization algorithms,” *arXiv preprint arXiv:1707.06347*, 2017.
- [31] V. Makoviychuk, L. Wawrzyniak, Y. Guo, M. Lu, K. Storey, M. Macklin, D. Hoeller, N. Rudin, A. Allshire, A. Handa, and G. State, “Isaac gym: High performance GPU based physics simulation for robot learning,” in *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021.
- [32] M. Chignoli, D. Kim, E. Stanger-Jones, and S. Kim, “The mit humanoid robot: Design, motion planning, and control for acrobatic behaviors,” in *2020 IEEE-RAS 20th International Conference on Humanoid Robots (Humanoids)*. IEEE, 2021, pp. 1–8.