DanceCamera3D: 3D Camera Movement Synthesis with Music and Dance

Zixuan Wang[1,2], Jia Jia[1,2*], Shikun Sun[1,2], Haozhe Wu[1,2], Rong Han[1], Zhenyu Li[1],
Di Tang[4], Jiaqing Zhou[4], Jiebo Luo[3*]
[1]Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China
[2]Beijing National Research Center for Information Science and Technology (BNRist)
[3]Department of Computer Science, University of Rochester, USA [4]ByteDance Hangzhou, China
{wangzixu21, ssk21, wuhz19, hanr21, zy-li21}@mails.tsinghua.edu.cn, jjia@tsinghua.edu.cn,
jluo@cs.rochester.edu, {minliu, jiashu}@bytedance.com

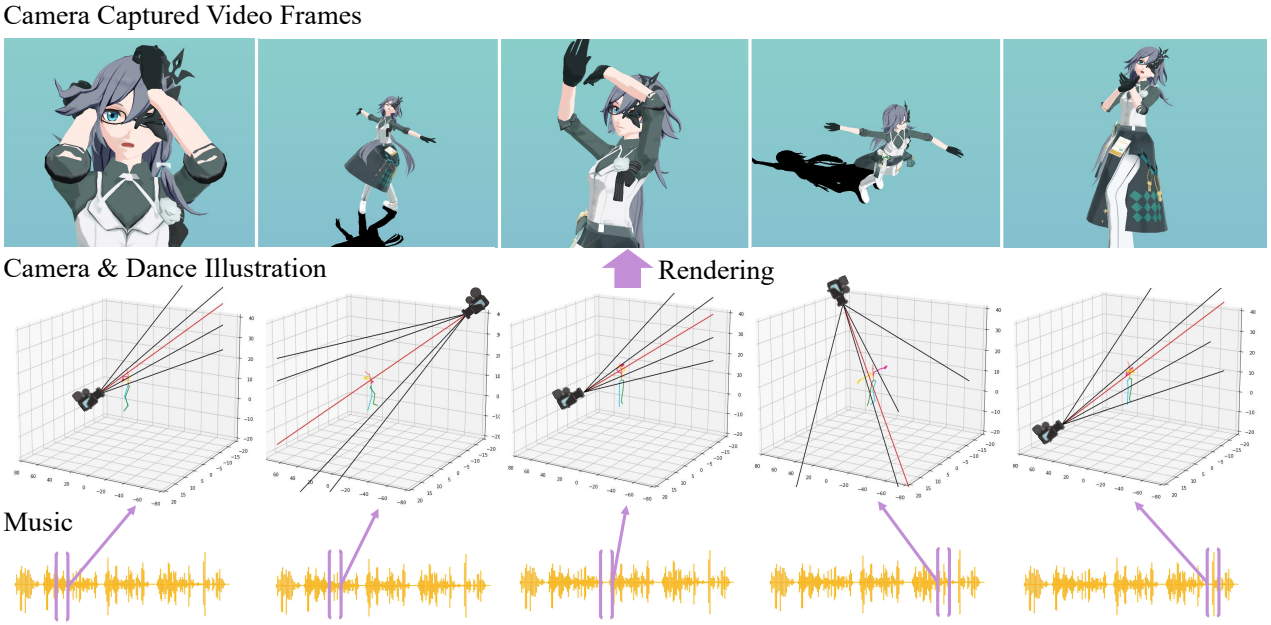


Figure 1. We present the **DCM** dataset, which contains 3.2 hours paired 3D **D**ance motion, **C**amera movement and **M**usic audio.

Abstract

*Choreographers determine what the dances look like, while cameramen determine the final presentation of dances. Recently, various methods and datasets have showcased the feasibility of dance synthesis. However, camera movement synthesis with music and dance remains an unsolved challenging problem due to the scarcity of paired data. Thus, we present **DCM**, a new multi-modal 3D dataset, which for the first time combines camera movement with dance motion and music audio. This dataset encompasses 108 dance sequences (3.2 hours) of paired dance-camera-music data from the anime community, covering 4 music genres. With this dataset, we uncover that dance camera movement is multifaceted and human-centric, and possesses multiple influencing factors, making dance camera synthesis a more challenging task compared to camera or dance synthesis alone. To overcome these difficulties, we propose **DanceCamera3D**, a transformer-based diffusion model that incorporates a novel body attention loss and a condition separation strategy. For evaluation, we devise new metrics measuring camera movement quality, diversity, and dancer fidelity. Utilizing these metrics, we conduct extensive experiments on our DCM dataset, providing both quantitative and qualitative evidence showcasing the effectiveness of our DanceCamera3D model. Code and video demos are available at https://github.com/Carmenw1203/DanceCamera3D-Official.*

1. Introduction

Dancing with the camera is a unique cinematic experience that merges cinematography and choreography. In the process of dance video production, moving the camera along with the dancer better captures the focus of dance motions

* Corresponding author

and provides the audience with a more immersive storytelling experience. However, dance camera movement is influenced by multifaceted factors including music and dance. Going further, a good dance camera should have diverse shot-type changes and human-centric characteristics. As a result, creating camera movement for dance is uniquely challenging. Meanwhile, existing methods [27, 36, 41] usually produce dance videos without camera movement which results in a boring single fixed view for the audience and uncontrollable situations where the dancer moves out of sight. Thus, how to automatically synthesize associated camera movement given music and dance is a significant and worth studying question.

Extensive works have made progress in music-dance dataset construction and synthesizing dance conditioned on music. However, camera movement generation with music and dance remains an open problem. This is mainly due to the following two main challenges:

– **Lack of Dance Camera Data.** Previous music-dance datasets acquire 2D data from dance videos or collect 3D data using the following three methods: motion capture (MoCap), 2D to 3D reconstruction, and animator edit. However, all previous datasets only focus on music and dance, or have difficulties capturing the mobile camera pose and trajectory. Specifically, MoCap-based methods [3, 40, 44, 52] rely on multi-camera with fixed positions or inertial sensors to achieve better accuracy so that it's harder and more complicated to use another mobile camera in the system and record the related parameters. Reconstruction-based methods [23, 27, 29, 38, 49] have problems extracting camera movements from dance video since it's confusing for the model to distinguish between camera movement and dancer movement. Animator-edited methods are suitable for camera movement design but previous related datasets [8, 26] only collect music audio and dance motion. In addition, some previous methods explore the camera movement extraction from 2D films, however relative positions of two characters are needed which is a very specific condition and cannot be applied to dance situations.
– **Complexity of Dance Cinematography.** Unlike dance choreography and normal cinematography, dance cinematography considers 1) multifaceted representation of camera movement including trajectory, direction, and field of view, 2) human-centric features denoting shot types and changes such as long shot, medium shot, close-up, cut-in, and cut-out, and 3) correlation with music and dance which indicates different moving speed, shot types, and body parts attention according to music and dance. Therefore, dance camera synthesis is more complicated than dance synthesis or normal cinematography.

To address the above issues, in this paper, we construct **DCM**, a new multi-model 3D **D**ance-**C**amera-**M**usic dataset, which for the first time collects camera keyframes and movements along with music and dance to advance the study of dance cinematography. We collect 108 dance sequences of paired dance-camera-music data from the anime community, which sum up to 3.2 hours and cover 4 languages of music.

With this dataset, we present **DanceCamera3D**, a transformer-based diffusion network, the first model that can robustly synthesize camera movement given music and dance. To better balance the effect of music and dance motion to camera movement, we propose a strong-weak condition separation strategy for classifier-free guidance (CFG) [16]. Meanwhile, we devise a new body attention loss to help DanceCamera3D achieve better focus on different limb parts. In addition, we introduce new metrics considering shot features and fidelity to the dancing character which are significant in dance cinematography. Using these new metrics and some rational metrics from dance synthesis, we conduct comprehensive quantitative and qualitative evaluations on our **DCM** dataset, which demonstrate that our DanceCamera3D outperforms the baseline models on quality, diversity, and fidelity. Experiments also approve that our strong-weak condition separation strategy helps the diffusion model acquire more feasibility in the trade-off among quality, diversity, and dancer fidelity. In summary, our contributions are as follows:

- We construct a new **DCM** dataset, which for the first time collects rich annotated camera data with multi-genre music and dance. Our DCM dataset possesses the potential to benefit the studies of dance camera synthesis, camera keyframing, and shot type classification.
- We introduce a novel Music Dance driven Camera Movement Synthesis task, which aims to automatically synthesize camera movement given music and dance. To our best knowledge, this is the first work that proposes and works on such a problem. To conduct comprehensive evaluations, we devise new metrics considering dance cinematography knowledge.
- We present **DanceCamera3D**, a transformer-based diffusion model, which is the first model for camera synthesis from music and dance. DanceCamera3D achieves more feasibility and better fidelity with a strong-weak condition separation strategy and a novel loss function.

2. Related Work

2.1. Dance and Camera Dataset

The construction of 2D and 3D music-dance datasets has attracted much attention since data-driven methods became popular in dance synthesis. Early works construct 2D music-dance datasets from videos. Authors of [24] extract 2D skeleton from dance videos using 2D pose estimation [6]. AIST [43] provides multi-view dancing videos

Dataset	Camera Data	Camera Keyframes	Camera Data Acquisition	Dance Data	Dance Keyframes	Dance Data Acquisition	Capture Environment
AIST [43]	Fixed-multi-view	✗	2D videos	2D	✗	2D videos	Lab-control
GrooveNet [3]	✗	✗	✗	3D	✗	MoCap	Lab-control
Dance with Melody [40]	✗	✗	✗	3D	✗	MoCap	Lab-control
FineDance [28]	✗	✗	✗	3D	✗	MoCap	Lab-control
AIST++ [27]	✗	✗	✗	3D	✗	Reconstruction	Lab-control
AIST-M [49]	✗	✗	✗	3D	✗	Reconstruction	Lab-control
AIOZ-GDANCE [23]	✗	✗	✗	3D	✗	Reconstruction	Lab-control
ChoreoMaster [8]	✗	✗	✗	3D	✗	Animator	In-the-wild
PhantomDance [26]	✗	✗	✗	3D	✓	Animator	In-the-wild
Jiang *et al*. [22]	Movable-view	✗	Reconstruction	✗	✗	✗	In-the-wild
Bonatti *et al*. [4]	Movable-view	✗	Defined-Rules	✗	✗	✗	Lab-control
Yu *et al*. [51]	Fixed-multi-view	✗	Animator	✗	✗	✗	Lab-control
DCM	Movable-view	✓	Animator	3D	✓	Animator	In-the-wild

Table 1. **Comparsion of dance-camera-music datasets**. Our DCM dataset is the first 3D dataset with dance, music, and camera movement including keyframe data, which can benefit the studies of dance camera synthesis, camera keyframing, and shot type classification.

paired with music. Meanwhile, tremendous progress has been made in the construction of 3D dance datasets. From the perspective of data acquisition methods, 3D dance datasets can be divided into three categories: motion capture (MoCap) based, reconstruction-based, and animator-edited datasets. Authors of [3, 28, 40, 44, 52] utilize MoCap to record 3D skeletons to build music-dance datasets. Considering the high cost for hiring dancers and equip devices of MoCap system, authors of [23, 27, 29, 38, 45, 49] propose to extract 3D dance pose from 2D dance video with tracking and pose estimation tools like AlphaPose [14] and MMPose [9]. Unlike the above two types of 3D datasets, animator-edited datasets [8, 26] are built from anime communities or hiring animators to annotate dance motions aligned with music. However, previous music-dance datasets all focus on music and dance data acquirement or have problems in capturing movable camera movement. Therefore, the camera correlation with music and dance is not exploited in these datasets. Besides, camera datasets are constructed for camera planning studies. Specifically, authors of [22] extract camera movement from 2D films, which relies on the positions of two characters. Authors of [4] produce multi-view tracking camera data with pre-defined movement rules in a photo-realistic simulator. Authors of [51] manually edit multi-camera with fixed positions for the study of camera placement in storytelling situations. Overall, existing camera datasets are limited in pre-defined rules or 2D estimation methods which have many constraints. Yet, movable camera data in dance situations is ignored. We compare different music-dance datasets in Table 1.

2.2. Dance Synthesis

Dance synthesis and dance camera synthesis are closely related problems since they share significant procedures including music feature extraction, encoding of dance motions and music features, and spatio-temporal forecasting. Extensive works have made progress in dance synthesis. Early methods [7, 13, 25, 30] regard music-to-dance as a similarity-based or statistical retrieval problem, which results in unrealistic choreographies and limited capacity. With the development of deep learning methods and large-scale datasets, researchers utilize neural networks to solve this problem. Typically, Crnkovic-Friis *et al*. [10] devise a Chor-RNN framework to predict dance motion. Tang *et al*. [40] synthesize dance motion using an LSTM-autoencoder. Wu *et al*. [46] employ Generative Adversarial Networks (GANs) to learn both music-to-dance and dance-to-music. Later, authors of [27, 36, 37] propose transformer-based methods auto-regressively, and authors of [41] use a diffusion model to synthesize dance in a denoising way. Meanwhile, some previous works [8, 50] introduce motion units from dance knowledge to produce more realistic dance, and some others [23, 45, 49] make efforts to generate group dance. However, previous approaches all focus on dance synthesis and ignore the significance of synthesizing camera movement in dance performance.

2.3. Camera Control and Planning

Automatic cinematography has attracted growing interest since manually producing film-like videos needs both professional practice and high labor but artistic video content is crucial in media entertainment and game industry. Jiang *et al*. [22] propose to extract camera behaviors from film clips and re-apply these behaviors in a virtual environment. Rao *et al*. [33] take story and camera scripts as input to compose dynamic storyboards with changing camera views in a virtual environment. Wu *et al*. [47] propose a GAN-based controller which generates actor-driven camera movements

considering spatial, emotional, and aesthetic factors. Rucks *et al*. [34] present CamerAI to imitate chase camera in third-person games which have viewpoints constraints. To produce better cutscenes in games, Evin *et al*. [12] devise CineAI by mimicking the cinematography techniques of movie directors. In addition, authors of [15, 18–20] make efforts in aerial cinematography studies which aim to automatically generate movement of camera drone with artistic principles and film style. Compared to previous problems, camera control and planning in dance is more challenging because the correlation of the camera movement with music and dance motions should be considered.

3. The DCM Dataset

3.1. Dataset Collection and Preprocessing

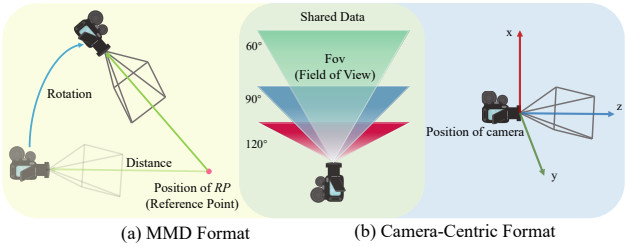
Since MoCap and reconstruction methods have difficulties capturing camera movement along with dance motion, we collect paired dance-camera-music data from the anime community. The raw data is MikuMikuDance (MMD) resources in which dance motions and camera movements are represented as keyframes with Bezier Curve parameters. However, the Bezier Curve makes it a non-differentiable process to calculate the in-between frames which is not suitable for back-propagation. Thus, for training with neural networks, we calculate motions for each joint and interpolate frames between keyframes with Bezier interpolation. Afterward, we align the dance, camera, and music data.

3.2. Dataset Description

After alignment, we have 108 pieces of paired data(193 minutes) covering music in 4 kinds of languages including English, Chinese, Korean, and Japanese. For camera pose representation, we originally acquired data in the MMD format. As shown in (a) of Figure 2, we assume RP is the reference point of camera pose, then the MMD format camera data includes the position of RP, rotation and distance relative to RP, and camera Fov(field of view). This format is not enough for training because it cannot directly reflect the spatial relation between camera and character, and absolute camera trajectory which are significant to dance camera synthesis. Thus, we calculate a Camera-Centric format which consists of global position, rotation, and Fov of the camera, as shown in (b) of Figure 2. Besides, dance motion in our data consists of rotations and positions of 60 joints. The FPS of dance motion and camera movement is 30.

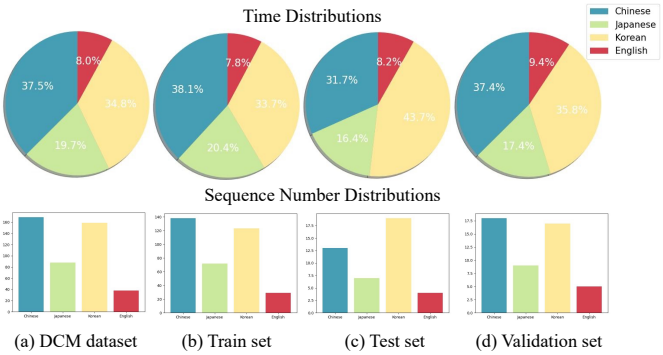
3.3. Dataset Split

The duration of the original data ranges from 17 to 267 seconds, so simply dividing them into the train, test, and validation sets cannot take both the music types and duration into account. To solve this problem, we first randomly cut our data into shorter pieces ranging from 17 to 35 seconds,



(a) MMD Format (b) Camera-Centric Format

Figure 2. **Camera pose formats in our DCM dataset.** (a) shows the original MMD format of camera pose including the position of RP, rotation and distance relative to RP, and Fov. (b) illustrates our Camera-Centric format consisting of the camera's Fov, global position, and rotation represented with x, y, and z vectors in the above figure.



(a) DCM dataset (b) Train set (c) Test set (d) Validation set

Figure 3. **Detailed distributions of our DCM dataset and split sets.**

in which all cut points are keyframes for better reservation of camera characteristics. Then for every music type, we randomly split the data with probabilities of 0.8 : 0.1 : 0.1 to obtain the train, test, and validation sets. As shown in Figure 3, we illustrate the detailed distributions of the split sets and our whole dataset.

4. Music & Dance Driven Camera Generation

4.1. Problem Formulation

The problem setting of music and dance conditioned camera generation is to predict the movement of the camera from given aligned music audio and dance poses. Here we represent music audio and dance pose conditions as $\boldsymbol{m} = \{m^1, m^2, \ldots, m^N\}$ and $\boldsymbol{p} = \{p^1, p^2, \ldots, p^N\}$ for a sequence with N frames. Since dance motion data in MMD resources have 60 frequently used joints, we represent dance pose with joints global positions: $p^i \in \mathbb{R}^{60\times 3}$. For the camera representation, we use the MMD format: $\boldsymbol{x} = \{x^1, x^2, \ldots, x^N\}$, $x^i \in \mathbb{R}^{3+3+1+1}$ for training, and calculate the camera-centric format: $\boldsymbol{xc} = \{xc^1, xc^2, \ldots, xc^N\}$, $xc^i \in \mathbb{R}^{3+3*3+1}$ for some loss functions. Overall, DanceCamera3D learns to predict camera $\boldsymbol{x}$ from input music $\boldsymbol{m}$ and dance $\boldsymbol{p}$.

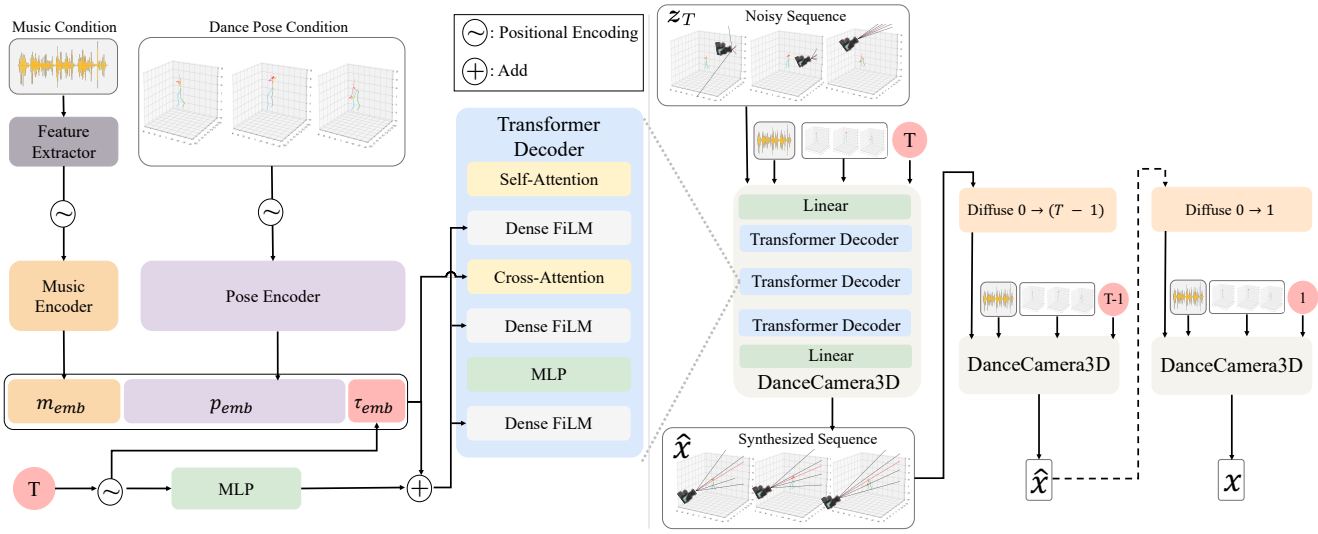


Figure 4. **Overview of DanceCamera3D Framework.** We adopt a transformer-based diffusion architecture to synthesize dance camera movement given music audio and dance pose as conditions. DanceCamera3D takes above conditions and a noisy sequence $\boldsymbol{z}_T \sim \mathcal{N}(0, \boldsymbol{I})$ as input and predicts noiseless sample $\hat{\boldsymbol{x}}$. Then we diffuse back $\hat{\boldsymbol{x}}$ and repeat the denoising process until $t = 0$ to acquire final results.

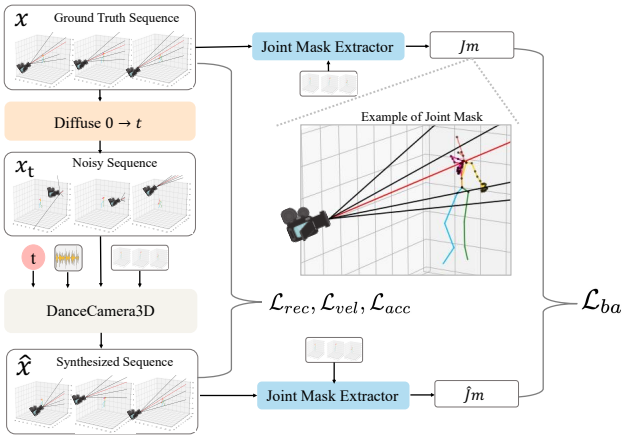


Figure 5. **Illustration of the training process and losses.** For each randomly sampled timestep t, we diffuse back the ground truth sequence to a noisy sequence. Then DanceCamera3D takes conditions, timestep, and a noisy sequence to predict camera movements $\hat{\boldsymbol{x}}$. We propose to detect joint masks indicating joints inside the camera view and devise the body attention loss $\mathcal{L}_{ba}$ based on joint masks which are represented with dots on the joints.

4.2. DanceCamera3D Architecture

As shown in Figure 4, DanceCamera3D uses a transformer-based diffusion model to synthesize camera movement in a denoising manner. Given music and dance pose conditions $\boldsymbol{m}$ and $\boldsymbol{p}$, we first extract the acoustic feature and then encode $\boldsymbol{m}$ and $\boldsymbol{p}$ to obtain music and pose embeddings $\boldsymbol{m}_{emb}$ and $\boldsymbol{p}_{emb}$. Then for generation process, we follow the DDPM [17] to define the diffusion as a Markov noising process with T steps, in which latents $\boldsymbol{x}_{t}^{T}{}_{t=0}$ follow a forward noising process $q(\boldsymbol{x}_t|\boldsymbol{x})$:

$$q(\boldsymbol{x}_t|\boldsymbol{x}) \sim \mathcal{N}(\sqrt{\bar{\alpha}_t}\boldsymbol{x}, (1-\bar{\alpha}_t)\boldsymbol{I}), \tag{1}$$

where $\boldsymbol{x} \sim p(\boldsymbol{x})$ is sampled from data distribution and $\bar{\alpha}_t \in (0, 1)$ are monotonically decreasing constants. In this way, we can approximately produce $\boldsymbol{x}_T \sim \mathcal{N}(0, \boldsymbol{I})$ when $\bar{\alpha}_t$ approaches 0. Reversely, our model learns to predict $\hat{\boldsymbol{x}}(\boldsymbol{x}_t, t, \boldsymbol{m}, \boldsymbol{p}) \approx \boldsymbol{x}$ for all t. Thus, our DanceCamera3D takes music, dance and a noisy sequence $\boldsymbol{z}_T \sim \mathcal{N}(0, \boldsymbol{I})$ as input to predict noiseless camera movement $\hat{\boldsymbol{x}}$. For inference, we noise $\hat{\boldsymbol{x}}$ back to timestep $t-1$ as $\boldsymbol{x}_{t-1}$ and repeat the denoising process until $t = 0$ to obtain final results.

4.3. Training and Losses

We illustrate the train process and losses for DanceCamera3D in Figure 5. Each time, we first randomly sample $t \in (0, T)$ and $\boldsymbol{x}$ from ground truth distribution. Then add noise for $\boldsymbol{x}$ to $\boldsymbol{x}_t$ with $q(\boldsymbol{x}_t|\boldsymbol{x})$. Afterward, we enter $\boldsymbol{m}$, $\boldsymbol{p}$, and t into the model and acquire synthesized camera movement sequence $\hat{\boldsymbol{x}}$. We train our model with commonly used classifier-free guidance (CFG) [16] in diffusion models. However, considering music and dance have quite different impacts on camera movement, we propose a strong-weak condition separation strategy to conduct CFG on these two conditions respectively instead of together, which is demonstrated to be effective in Sec 5.4. So far, we can restrict the synthesized results to comply with the conditions using losses. For physical realism, we select commonly used reconstruction loss $\mathcal{L}_{rec}$, velocity loss $\mathcal{L}_{vel}$ and

acceleration loss $\mathcal{L}_{acc}$:

$$\begin{aligned} \mathcal{L}_{rec} &= ||\boldsymbol{x} - \hat{\boldsymbol{x}}||_2^2, \\ \mathcal{L}_{vel} &= ||\boldsymbol{x}' - \hat{\boldsymbol{x}}'||_2^2, \\ \mathcal{L}_{acc} &= ||\boldsymbol{x}'' - \hat{\boldsymbol{x}}''||_2^2, \end{aligned} \tag{2}$$

where $\boldsymbol{x}'$ and $\boldsymbol{x}''$ represent the first-order (velocity) and second-order (acceleration) partial derivatives of camera movement parameters $\boldsymbol{x}$ on time. However, these commonly used losses for movement synthesis cannot help the model to capture the significance of the dancer's motion and even move the dancer out of camera view, for which we provide a more detailed discussion in Sec 5.5. To solve this problem, we propose a body attention loss $\mathcal{L}_{ba}$:

$$\mathcal{L}_{ba} = ||\boldsymbol{Jm} - \hat{\boldsymbol{Jm}} * \boldsymbol{Jm}||, \tag{3}$$

where $\boldsymbol{Jm}$ denotes whether joints are inside or outside the camera view:

$$\boldsymbol{Jm}_j^i = \begin{cases} 1 & p_j^i \text{ inside Camera View,} \\ 0 & p_j^i \text{ outside Camera View,} \end{cases} \tag{4}$$

where p_j^i refers to position of joint j at frame i. For better visualization of the bone mask, we show a sample with joint dots in Figure 5. More details on the implementation of $\mathcal{L}_{ba}$ and $\boldsymbol{Jm}$ are illustrated in supplementary materials. Using $\mathcal{L}_{ba}$, the model is restricted to concentrate more on significant body parts that are captured in ground truth. Our overall training loss consists of a weighted sum of the above losses, while $\lambda_{vel}, \lambda_{acc}, \lambda_{ba}$ are trade-off weights:

$$\mathcal{L} = \mathcal{L}_{rec} + \lambda_{vel}\mathcal{L}_{vel} + \lambda_{acc}\mathcal{L}_{acc} + \lambda_{ba}\mathcal{L}_{ba}. \tag{5}$$

5. Experiments

5.1. Experimental Setup

Dataset Preparation. All the experiments in this paper are conducted on our DCM dataset which for the first time collects camera movement with dance and music. As mentioned in Sec 3.3, we split DCM into train, test, and validation sets, and report the performance on the test set only. For the training of DanceCamera3D, we split the train data into 5-second subsequences with a stride of 0.5 seconds.

Implementation Details. In our experiment, the input of the model is aligned dance motion and music audio except for a transformer baseline needs an extra 2.5-second (75 frames) camera seed movements. The output of the model is camera movement sequences with the same length to input dance and music. During inference, we generate 5-second subsequences with a stride of 2.5 seconds, then we interpolate the overlapping slices to enforce consistency with linear decaying weight. Afterward, we use a total variation denoiser [11] to detect the keyframes in our results and use a Savitzky-Golay filter [35] to smooth camera movements between keyframes. For the training process, all our experiments are trained with 128 batch size for 3000 epochs using Adan [48] optimizer. Our final model has 52.7M total parameters, which was trained on 6 NVIDIA 3090 GPUs for 13 hours. We utilize "Jukebox" to extract music features that have 4800 dimensions information for each frame. For diffusion timesteps, we use $T = 1000$.

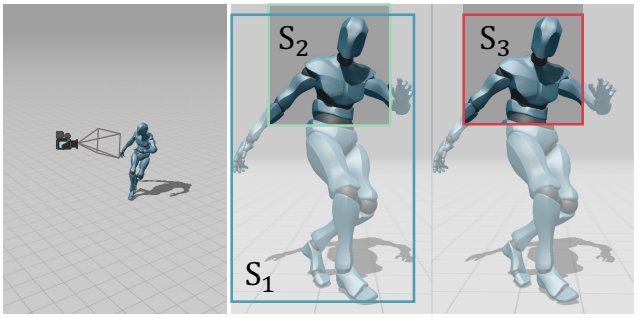


Figure 6. **Significant factors for shot features.** S_1 is the area of the dancer projected onto the camera plane, S_2 is the camera screen area and S_3 is the area of the dancer's body parts inside the camera screen. Here we use the character model from Mixamo [2].

5.2. Evaluation Metrics

Kinetic Feature Evaluation. Following prior works [27, 36], we evaluate generated camera movement using Frechet Inception Distance (FID) for quality and average Euclidean distance (Dist) in the feature space for diversity. For kinetic evaluation, we use a kinetic feature extractor [31] following existing works [27, 36]. Since this feature extractor calculates average velocity and acceleration, we compute kinetic features on split 2.5-second data to ensure the density of feature distribution which is similar to settings in AIST++ [27]. Thus, we have got FID_k for kinetic quality and Dist_k for kinetic diversity.

Shot Feature Evaluation. Shot features are significant to dance camera synthesis, however existing works [5, 32, 42] are limited to 2D classifications with finite shot types. So we newly devise a shot feature extractor in 3D scenes, considering the knowledge of cinematography. As shown in Figure 6, we calculate shot features as:

$$\text{Features}_{shot} = (S_3/S_1, S_3/S_2). \tag{6}$$

where S_1 and S_3 indicate camera plane projection areas of the dancer's whole body and body parts inside the camera screen. S_2 is the camera screen area. Using this formulation, S_3/S_1 represents the percentage of the body inside the camera view and S_3/S_2 denotes the proportion of the camera screen that the dancer occupies. Then, we calculate FID and Dist for Features_{shot} and its velocity to get FID_s and

Method	Quality		Diversity		Dancer Fidelity		User Study
	$FID_k \downarrow$	$FID_s \downarrow$	$Dist_k \uparrow$	$Dist_s \uparrow$	DMR$\downarrow$	LCD$\downarrow$	DanceCamera3D WinRate$\uparrow$
Ground Truth	-	-	3.275	1.731	0.00142	-	37.62% $\pm$ 2.83%
DanceRevolution* [21]	10.267	2.368	1.491	1.118	0.0062	0.154	71.90% $\pm$ 2.38%
FACT# [27]	5.205	0.960	1.505	1.007	0.0070	0.151	65.71% $\pm$ 1.71%
DanceCamera3D w/o $\mathcal{L}_{ba}$	4.022	0.728	1.421	**1.671**	0.0899	0.310	77.14% $\pm$ 3.53%
DanceCamera3D (Ours)	**3.749**	**0.280**	**1.631**	1.326	**0.0025**	**0.147**	-

Table 2. **Comparison of our DanceCamera3D with Spatio-Temporal Models.** * means we utilize the LSTM decoder of DanceRevolution [21] to generate camera motion frame by frame. # means we follow FACT [27] to autoregressively synthesize camera motion with seed motions. - denotes that the self-comparison is meaningless.

$Dist_s$ for shot quality and diversity. Considering the difference between shot and kinetic features, we compute shot metrics frame-by-frame to keep the accuracy of shot types.

Dancer Fidelity Evaluation. Dancer fidelity means camera movement should try to capture significant body parts against the dancer's poses and avoid the long time absence of the dancer in the camera view. We propose to evaluate dancer fidelity with the following two metrics: 1) Dancer Missing Rate (DMR): DMR represents the ratio of frames in which the dancer is not in the view of the camera, and 2) Limbs Capture Difference (LCD): LCD denotes the difference of body parts inside and outside camera view between synthesized results and ground truth. Lower DMR and LCD mean better dancer fidelity for fewer dancer-missing situations and more similarity between results and ground truth which is carefully modified.

User Study. For qualitative evaluation, we conduct a user study to compare our method with baseline methods, ablation method, and ground truth. In this study, we first randomly select 10 dance-camera inputs from the test set ranging from 17$\sim$35 seconds and sample results from our methods and compared methods. For each baseline result, we combine the corresponding results from our method and produce 40 pairs of dance videos. Then, we invite 21 participants to watch all these 40 pairs of videos in a random shuffled order and answer the question "Which camera movement better showcases the dance and music? LEFT or RIGHT" for each pair of videos. The invited 21 participants consist of dancers, animators, filmmakers, and people who have rare expertise with camera and dance.

5.3. Comparison with Spatio-Temporal Models

Since there is no existing method for music-dance conditioned camera synthesis, we implement some baselines following an auto-regressive generation scheme which has achieved strong qualitative performance in dance synthesis:

- **DanceRevolution [21].** Following DanceRevolution, we synthesize camera movement autoregressively with a 3-layer LSTM decoder.
- **FACT [27].** Following FACT, we use a transformer decoder to generate camera movement with 2.5-second seed movements in an autoregressive scheme. Bailando [36] also achieves strong qualitative performance using transformers, but they pre-train dance motions as motion units which cannot be applied to the camera since camera movements have a strong correlation with dancers' positions so it's hard to encode them as independent units.

For comparison fairness, we utilize the same feature encoders and losses with our model. Results are shown in Table 2, which demonstrate that our DanceCamere3D achieves better quality and higher diversity on both kinetic and shot features while preserving more dancer fidelity. The user study shows our method surpasses baseline methods by at least 65.71% winning rate. Compared to ground truth camera movements, our synthesized camera movements still have 37.62%. Feedback from users tells us that our model produces satisfying camera movements with considerable shot-type changes and a quite good focus on the character, but ground truth movements have more granular control and fewer artifacts since they are manually edited by animators. The case study also shows that our DanceCamera3D surpasses baseline methods and produces competitive results against ground truth, as illustrated in Figure 8.

5.4. Comparison on CFG

Classifier-free guidance (CFG) has been demonstrated to achieve state-of-the-art results for image generation [16, 39] and dance synthesis [41] using explicit control over the diversity-fidelity trade-off. However, dance camera synthesis is a more complex situation, since this problem has 2 conditions in which dance motion is strongly correlated to the camera and music audio is weakly correlated. Thus, we devise a strategy to separate dance and music conditions and conduct experiments on it. As shown in Figure 7, we illustrate results with different guidance weights ω_1, ω_2 to dance and music conditions including: 1) Red lines: $\omega_1 = \omega_2 = \omega$, 2) Blue lines: $\omega_1 = \omega + 0.25$, $\omega_2 = \omega$ and 3) Green lines: $\omega_1 = \omega$, $\omega_2 = \omega + 0.25$. Overall, results demonstrate that CFG strengthens the diversity and quality of camera movement while trading off dancer fidelity. Notably, too large guidance weights cause a drop in the quality of camera movement. This is because

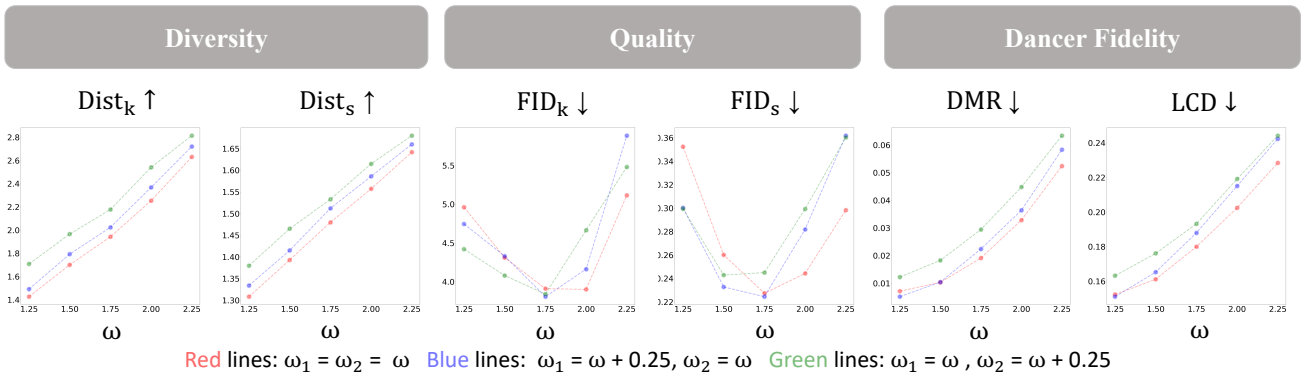


Figure 7. **Comparison of condition separation strategy on CFG.** Red lines show the results of applying equal guidance weights ω_1, ω_2. Based on this, we separately add 0.25 guidance weight on ω_1 and ω_2, indicating enhancements in dance and music conditions which are represented with Blue and Green lines. Overall, CFG strengthens the diversity and quality of camera movement by trading off dancer fidelity. Compared to equal guidance on all conditions, adding guidance separately allows more fine-grained control of the trade-offs.

overdose enhancement on conditions will move the results away from the ground truth distribution. Comparing green and blue lines, we find the music condition produces a more intense effect on camera movements, and the dance motion condition causes slower changes and better quality for achieving lower FID$_k$ and FID$_s$ with less loss on dancer fidelity at some points. This complies with the reality that, dance motion as a strong condition brings more focus on dancers, and music as a weak condition influences more on movement style. In summary, our strong-weak conditions separation strategy provides more fine-grained control on the trade-offs in dance camera synthesis.

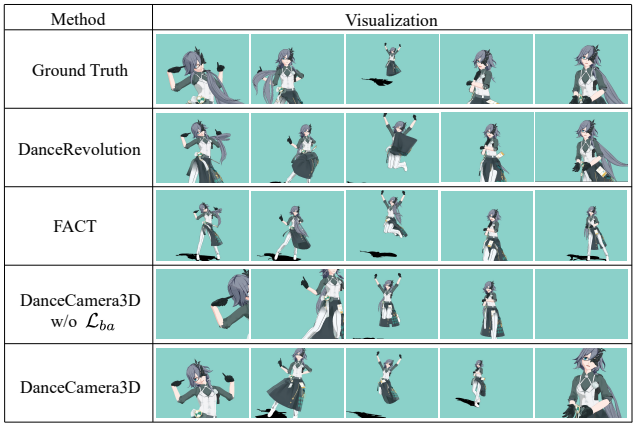


Figure 8. Visual comparison of rendered dance videos with synthesized camera movement from our DanceCamera3D and baseline methods. Compared to DanceRevolution and FACT, our DanceCamera3D produces more shot-type changes. DanceCamera3D w/o $\mathcal{L}_{ba}$ produces unbearable artifacts of failing to capture the position of the dancer which proves the effectiveness of $\mathcal{L}_{ba}$. Here we use the character model from [1].

5.5. Ablation Study

We conduct an ablation experiment to study the effectiveness of our newly designed body attention loss $\mathcal{L}_{ba}$. As shown in Table 2, quantitative evaluations show that the quality, dancer fidelity, and kinetic diversity decrease when we remove the $\mathcal{L}_{ba}$. The diversity of shot increases since there are more frames without dancer in the camera screen, which greatly change the distribution of shot features. User study also proves that model with $\mathcal{L}_{ba}$ produces more stable results with fewer artifacts like failing to capture the dancer or placing the dancer at the edge of the screen for a long time. As shown in Figure 8, the model without $\mathcal{L}_{ba}$ is more likely to generate unbearable artifacts which demonstrate the effectiveness of $\mathcal{L}_{ba}$.

6. Conclusion and Future Work

In this paper, we introduce a novel and valuable task: Music Dance driven Camera Movement Synthesis. To address this challenging problem, we constructed a new dataset DCM, which for the first time simultaneously collects camera, dance, and music data together with rich annotations. With this dataset, we present DanceCamera3D with a novel loss function and condition separation strategy, that can synthesize high-quality 3D dance camera movement given music and dance. We conduct comprehensive evaluations on the DCM dataset with newly devised metrics. Through extensive experiments, we demonstrate the effectiveness of our DanceCamera3D. To encourage further research in the fields of music, choreography, and cinematography, we will make both the source code and the dataset openly available as open-source resources. We believe our DCM dataset can not only facilitate the studies of dance camera synthesis but also contribute to research like camera keyframing and shot type classification.

7. Acknowledgements

This work is supported by the National Key R&D Program of China under Grant No.2021QY1500.

References

[1] Model modified and provided by https://space.bilibili.com/1561923759, character copyrights to miHoYo https://www.mihoyo.com. 8

[2] https://www.mixamo.com/, Mixamo. 6

[3] Omid Alemi, Jules Françoise, and Philippe Pasquier. Groovenet: Real-time music-driven dance movement generation using artificial neural networks. *networks*, 8(17):26, 2017. 2, 3

[4] Rogerio Bonatti, Arthur Bucker, Sebastian Scherer, Mustafa Mukadam, and Jessica Hodgins. Batteries, camera, action! learning a semantic control space for expressive robot cinematography. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7302–7308. IEEE, 2021. 3

[5] Luca Canini, Sergio Benini, and Riccardo Leonardi. Classifying cinematographic shot types. *Multimedia tools and applications*, 62:51–73, 2013. 6

[6] 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 Transactions on Pattern Analysis and Machine Intelligence*, 2019. 2

[7] Marc Cardle, Loic Barthe, Stephen Brooks, and Peter Robinson. Music-driven motion editing: Local motion transformations guided by music analysis. In *Proceedings 20th Eurographics UK Conference*, pages 38–44. IEEE, 2002. 3

[8] Kang Chen, Zhipeng Tan, Jin Lei, Song-Hai Zhang, Yuan-Chen Guo, Weidong Zhang, and Shi-Min Hu. Choreomaster: choreography-oriented music-driven dance synthesis. *ACM Transactions on Graphics (TOG)*, 40(4):1–13, 2021. 2, 3

[9] MMPose Contributors. Openmmlab pose estimation toolbox and benchmark. https://github.com/open-mmlab/mmpose, 2020. 3

[10] Luka Crnkovic-Friis and Louise Crnkovic-Friis. Generative choreography using deep learning. *arXiv preprint arXiv:1605.06921*, 2016. 3

[11] Huiqian Du and Yilin Liu. Minmax-concave total variation denoising. *Signal, Image and Video Processing*, 12:1027–1034, 2018. 6

[12] Inan Evin, Perttu Hämäläinen, and Christian Guckelsberger. Cine-ai: Generating video game cutscenes in the style of human directors. *Proceedings of the ACM on Human-Computer Interaction*, 6(CHI PLAY):1–23, 2022. 4

[13] Rukun Fan, Songhua Xu, and Weidong Geng. Example-based automatic music-driven conventional dance motion synthesis. *IEEE transactions on visualization and computer graphics*, 18(3):501–515, 2011. 3

[14] Hao-Shu Fang, Jiefeng Li, Hongyang Tang, Chao Xu, Haoyi Zhu, Yuliang Xiu, Yong-Lu Li, and Cewu Lu. Alphapose: Whole-body regional multi-person pose estimation and tracking in real-time. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. 3

[15] Mirko Gschwindt, Efe Camci, Rogerio Bonatti, Wenshan Wang, Erdal Kayacan, and Sebastian Scherer. Can a robot become a movie director? learning artistic principles for aerial cinematography. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1107–1114, 2019. 4

[16] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. 2, 5, 7

[17] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 5

[18] Chong Huang, Chuan-En Lin, Zhenyu Yang, Yan Kong, Peng Chen, Xin Yang, and Kwang-Ting Cheng. Learning to film from professional human motion videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4244–4253, 2019. 4

[19] Chong Huang, Zhenyu Yang, Yan Kong, Peng Chen, Xin Yang, and Kwang-Ting Tim Cheng. Learning to capture a film-look video with a camera drone. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 1871–1877, 2019.

[20] Chong Huang, Yuanjie Dang, Peng Chen, Xin Yang, and Kwang-Ting Cheng. One-shot imitation drone filming of human motion videos. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(9):5335–5348, 2022. 4

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

[22] Hongda Jiang, Bin Wang, Xi Wang, Marc Christie, and Baoquan Chen. Example-driven virtual cinematography by learning camera behaviors. *ACM Transactions on Graphics (TOG)*, 39(4):45–1, 2020. 3

[23] Nhat Le, Thang Pham, Tuong Do, Erman Tjiputra, Quang D Tran, and Anh Nguyen. Music-driven group choreography. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8673–8682, 2023. 2, 3

[24] Hsin-Ying Lee, Xiaodong Yang, Ming-Yu Liu, Ting-Chun Wang, Yu-Ding Lu, Ming-Hsuan Yang, and Jan Kautz. Dancing to music. *Advances in neural information processing systems*, 32, 2019. 2

[25] Minho Lee, Kyogu Lee, and Jaeheung Park. Music similarity-based approach to generating dance motion sequence. *Multimedia tools and applications*, 62:895–912, 2013. 3

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

[27] Ruilong Li, Shan Yang, David A Ross, and Angjoo Kanazawa. Ai choreographer: Music conditioned 3d dance generation with aist++. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13401–13412, 2021. 2, 3, 6, 7

[28] Ronghui Li, Junfan Zhao, Yachao Zhang, Mingyang Su, Zeping Ren, Han Zhang, Yansong Tang, and Xiu Li. Finedance: A fine-grained choreography dataset for 3d full body dance generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10234–10243, 2023. 3

[29] Davide Moltisanti, Jinyi Wu, Bo Dai, and Chen Change Loy. Brace: The breakdancing competition dataset for dance motion synthesis. In *European Conference on Computer Vision*, pages 329–344. Springer, 2022. 2, 3

[30] Ferda Ofli, Engin Erzin, Yücel Yemez, and A Murat Tekalp. Learn2dance: Learning statistical music-to-dance mappings for choreography synthesis. *IEEE Transactions on Multimedia*, 14(3):747–759, 2011. 3

[31] Kensuke Onuma, Christos Faloutsos, and Jessica K Hodgins. Fmdistance: A fast and effective distance function for motion capture data. In *Eurographics (Short Papers)*, pages 83–86, 2008. 6

[32] Anyi Rao, Jiaze Wang, Linning Xu, Xuekun Jiang, Qingqiu Huang, Bolei Zhou, and Dahua Lin. A unified framework for shot type classification based on subject centric lens. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16*, pages 17–34. Springer, 2020. 6

[33] Anyi Rao, Xuekun Jiang, Yuwei Guo, Linning Xu, Lei Yang, Libiao Jin, Dahua Lin, and Bo Dai. Dynamic storyboard generation in an engine-based virtual environment for video production. In *ACM SIGGRAPH 2023 Posters*, pages 1–2. 2023. 3

[34] James Rucks and Nikolaos Katzakis. Camerai: Chase camera in a dense environment using a proximal policy optimization-trained neural network. In *2021 IEEE Conference on Games (CoG)*, pages 1–8. IEEE, 2021. 4

[35] Ronald W Schafer. What is a savitzky-golay filter?[lecture notes]. *IEEE Signal processing magazine*, 28(4):111–117, 2011. 6

[36] Li Siyao, Weijiang Yu, Tianpei Gu, Chunze Lin, Quan Wang, Chen Qian, Chen Change Loy, and Ziwei Liu. Bailando: 3d dance generation by actor-critic gpt with choreographic memory. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11050–11059, 2022. 2, 3, 6, 7

[37] Li Siyao, Weijiang Yu, Tianpei Gu, Chunze Lin, Quan Wang, Chen Qian, Chen Change Loy, and Ziwei Liu. Bailando++: 3d dance gpt with choreographic memory. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023. 3

[38] Guofei Sun, Yongkang Wong, Zhiyong Cheng, Mohan S Kankanhalli, Weidong Geng, and Xiangdong Li. Deepdance: music-to-dance motion choreography with adversarial learning. *IEEE Transactions on Multimedia*, 23:497–509, 2020. 2, 3

[39] Shikun Sun, Longhui Wei, Zhicai Wang, Zixuan Wang, Junliang Xing, Jia Jia, and Qi Tian. Inner classifier-free guidance and its taylor expansion for diffusion models. In *The Twelfth International Conference on Learning Representations*, 2023. 7

[40] Taoran Tang, Jia Jia, and Hanyang Mao. Dance with melody: An lstm-autoencoder approach to music-oriented dance synthesis. In *Proceedings of the 26th ACM international conference on Multimedia*, pages 1598–1606, 2018. 2, 3

[41] Jonathan Tseng, Rodrigo Castellon, and Karen Liu. Edge: Editable dance generation from music. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 448–458, 2023. 2, 3, 7

[42] Ioannis Tsingalis, Nicholas Vretos, Nikos Nikolaidis, and Ioannis Pitas. Svm-based shot type classification of movie content. In *Mediterranean Electrotechnical Conference*, 2012. 6

[43] 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 *ISMIR*, page 6, 2019. 2, 3

[44] Guillermo Valle-Pérez, Gustav Eje Henter, Jonas Beskow, Andre Holzapfel, Pierre-Yves Oudeyer, and Simon Alexanderson. Transflower: probabilistic autoregressive dance generation with multimodal attention. *ACM Transactions on Graphics (TOG)*, 40(6):1–14, 2021. 2, 3

[45] Zixuan Wang, Jia Jia, Haozhe Wu, Junliang Xing, Jinghe Cai, Fanbo Meng, Guowen Chen, and Yanfeng Wang. Groupdancer: Music to multi-people dance synthesis with style collaboration. In *Proceedings of the 30th ACM International Conference on Multimedia (MM)*, pages 1138–1146, 2022. 3

[46] Shuang Wu, Zhenguang Liu, Shijian Lu, and Li Cheng. Dual learning music composition and dance choreography. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 3746–3754, 2021. 3

[47] Xinyi Wu, Haohong Wang, and Aggelos K Katsaggelos. The secret of immersion: actor driven camera movement generation for auto-cinematography. *arXiv preprint arXiv:2303.17041*, 2023. 3

[48] Xingyu Xie, Pan Zhou, Huan Li, Zhouchen Lin, and Shuicheng Yan. Adan: Adaptive nesterov momentum algorithm for faster optimizing deep models. *arXiv preprint arXiv:2208.06677*, 2022. 6

[49] Siyue Yao, Mingjie Sun, Bingliang Li, Fengyu Yang, Junle Wang, and Ruimao Zhang. Dance with you: The diversity controllable dancer generation via diffusion models. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 8504–8514, 2023. 2, 3

[50] Zijie Ye, Haozhe Wu, Jia Jia, Yaohua Bu, Wei Chen, Fanbo Meng, and Yanfeng Wang. Choreonet: Towards music to dance synthesis with choreographic action unit. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 744–752, 2020. 3

[51] Zixiao Yu, Enhao Guo, Haohong Wang, and Jian Ren. Bridging script and animation utilizing a new automatic cinematography model. In *2022 IEEE 5th International Conference on Multimedia Information Processing and Retrieval (MIPR)*, pages 268–273. IEEE, 2022. 3

[52] Wenlin Zhuang, Congyi Wang, Jinxiang Chai, Yangang Wang, Ming Shao, and Siyu Xia. Music2dance: Dancenet for music-driven dance generation. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 18(2):1–21, 2022. 2, 3

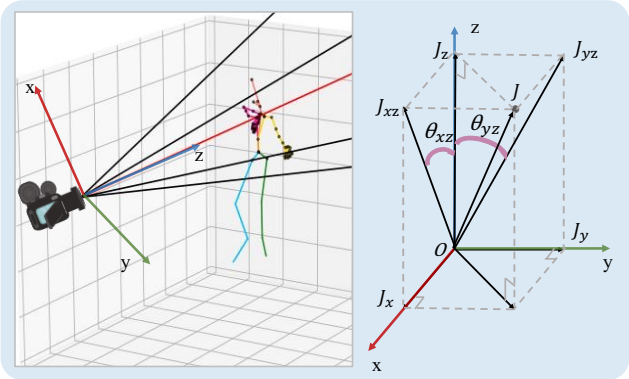
Appendix

A. Calculation of Joint Mask

As shown in Figure 9, we illustrate camera coordinates in (a) Global View and (b) Camera Local View. Given that O

Music Languages	Total Time	BPM		Camera Keyframes Interval		Number of Sequences		Frames of Sequences	
		Range	Average	Range	Average	Aligned	Split	Aligned	Split
Chinese	4298.8s	77.8∼143.6	119.2	1∼475	18.45	38	169	524∼8025	510∼1051
Japanese	2258.8s	86.1∼161.5	129.4	1∼220	6.67	15	88	1618∼7290	512∼1046
Korean	3996.5s	86.1∼161.5	119.8	1∼231	11.93	40	159	550∼6260	516∼1042
English	916.6s	99.4∼143.6	122.6	1∼228	18.57	15	38	829∼5737	510∼1046

Table 3. **Detailed statistics of the DCM dataset.** 'Aligned' means data after alignment among dance, camera, and music. 'Split' denotes data split into subsequences within 17∼35s which is more suitable for training.



(a) Global View (b) Camera Local View

Figure 9. **Details of joint mask calculation.**

is the camera eye at frame i and J is an arbitrary joint at frame i, our target is to determine whether J is inside the camera view or not. Specifically, we first project J onto the xz-plane and the yz-plane to get J_{xz} and J_{yz}, as shown in (b) of Figure 9. Supposing that joint masks of joints inside camera view are 1 and others are 0, we can represent the joint mask of J as Jm:

$$Jm = \begin{cases} 1 & \theta_{xz} \leq Fov/2, \theta_{yz} \leq Fov/2, \\ 0 & \text{others}, \end{cases} \tag{7}$$

where θ_{xz} is the angle between $\overrightarrow{OJ}_{xz}$ and $\overrightarrow{OJ}_z$, θ_{yz} is the angle between $\overrightarrow{OJ}_{yz}$ and $\overrightarrow{OJ}_z$, and Fov is the field of camera view at frame i. In more detail, we use the cosine function to compare angles, because the cosine function is symmetrical about 0 and monotonically decreases on $[0, \pi)$. Thus, we can further represent Jm as:

$$Jm = \begin{cases} 1 & Cos(\theta_{xz}) \geq Cos(Fov/2), Cos(\theta_{yz}) \geq Cos(Fov/2), \\ 0 & \text{others}, \end{cases} \tag{8}$$

For computing $Cos(\theta_{xz})$ and $Cos(\theta_{yz})$, we take $Cos(\theta_{xz})$ as an example:

$$\begin{aligned} Cos(\theta_{xz}) &= \frac{\overrightarrow{OJ}_{xz} * \overrightarrow{z_0}}{||\overrightarrow{OJ}_{xz}|| * ||\overrightarrow{z_0}||}, \\ \overrightarrow{OJ}_{xz} &= \overrightarrow{OJ} - \overrightarrow{OJ}_y, \\ \overrightarrow{OJ}_y &= (\overrightarrow{OJ} * \overrightarrow{y_0})\overrightarrow{y_0}, \end{aligned} \tag{9}$$

where $\overrightarrow{x_0}, \overrightarrow{y_0}, \overrightarrow{z_0}$ are unit vectors of x, y, z axes. Here $\overrightarrow{x_0}, \overrightarrow{y_0}, \overrightarrow{z_0}$ and the position of O make up the camera-centric format representation xc, which is mentioned in Sec 3.2 and Sec 4.1 of the full paper.

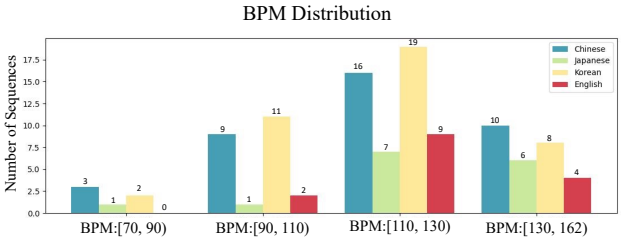


Figure 10. **BPM Distribution of DCM Dataset.**

B. Implementation of Body Attention Loss

In Sec 4.3, we represent our body attention loss $\mathcal{L}_{ba}$ as:

$$\mathcal{L}_{ba} = ||\boldsymbol{Jm} - \hat{\boldsymbol{Jm}} * \boldsymbol{Jm}||, \tag{10}$$

where $\hat{\boldsymbol{Jm}}$ means the generated joint mask and $\boldsymbol{Jm}$ means the ground-truth joint mask. This concise and clear representation denotes that we penalize the joints that are inside the camera view in ground truth but outside the camera view in synthesized results. However, in the actual implementation of $\mathcal{L}_{ba}$, we find that the calculation of joint mask is underivable. Thus, we implement $\mathcal{L}_{ba}$ as:

$$\begin{aligned} \mathcal{L}_{ba} = &\text{Relu}(\boldsymbol{Jm} * (\text{Cos}(\frac{Fov}{2}) - \text{Cos}(\boldsymbol{\theta_{xz}}))) \\ &+ \text{Relu}(\boldsymbol{Jm} * (\text{Cos}(\frac{Fov}{2}) - \text{Cos}(\boldsymbol{\theta_{yz}}))) \end{aligned} \tag{11}$$

where $\frac{Fov}{2}$ indicates the vector of camera field of view, $\boldsymbol{\theta_{xz}}$ and $\boldsymbol{\theta_{yz}}$ denote vectors of θ_{xz} and θ_{yz} respectively. In this way, for the joint outside the camera field of view in ground truth, the Jm is 0 and the corresponding impact to $\mathcal{L}_{ba}$ is 0. For the joint inside the camera field of view in ground truth, the Jm is 1, and the corresponding impact to $\mathcal{L}_{ba}$ is 0 only if this joint is inside the camera field of view in the generated result. Thus, this loss realizes a similar penalty as Equation 10.

C. Details of DCM Dataset

C.1. BPM

The BPMs (beat per minute) of music pieces in our DCM dataset range from 71 to 162. In detail, we illustrate BPM distribution with music categories in Figure 10.

C.2. Detailed Statistics

As shown in Table 3, we present more detailed statistics of our DCM dataset.