A. Details of the Training Process

As shown in the figure below. We trained the two stages separately to save graphics memory. The Global Diffusion is trained on long music input and sparse key motions extracted from ground truth. The output key motions of Global Diffusion are categories in $\boldsymbol{d_h}$ and $\boldsymbol{d_s}$ to guide the Local Diffusion only in the inference phase.

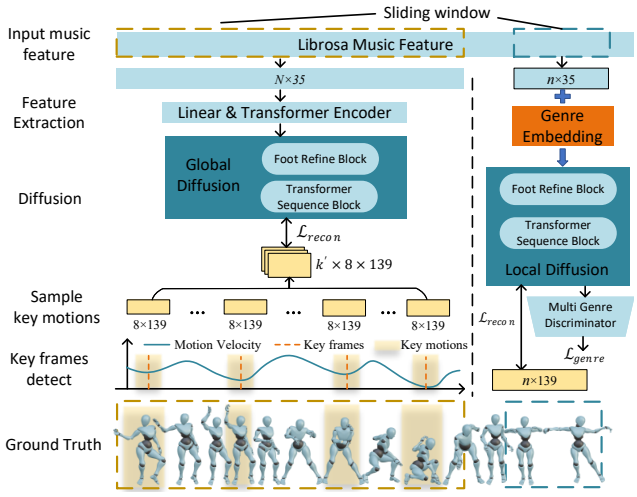

Figure 1. The Training process of Lodge.

B. Details of the Hard/Soft Diffusion Guidance

We categorize the characteristic dance primitives generated by global diffusion into hard-cue key motions d_h and soft-cue key motions d_s. We employ distinct diffusion guidance strategies for each, enabling them to guide local diffusion.

The role of d_h is to guide the local diffusion in generating the initial and final segments of the dance, ensuring that the concurrently generated dance fragments can seamlessly concatenate into a coherent, long-form dance. Therefore, we adopt Hard Diffusion Guidance for this purpose.

On the other hand, d_s serves to provide guidance to local diffusion. In this case, we aim for the guidance to be flexible, avoiding any disruption to the coherence of the dance generated by local diffusion. Consequently, we propose the Soft Diffusion Guidance algorithm for d_s. As illustrated in the pseudocode below, our proposed soft diffusion operates only for the first $1000 \times (1 - s)$ steps, where s is a hyperparameter. The impact of different s values on the results is detailed in Table 3 of the main paper.

```
import torch, librosa
# m is the given music feature, m.shape = [L,
    35], L is the time length
m = m[:ln] # l = L//n, n is the output frame
    number of one local diffusion
d_h, d_s = GlobalDiffusion(m)
# d_h.shape = [(l+1),8,139]; d_s.shape = [2l
    ,8,139]
d_h = d_h.reshape([(l+1)*8,139])
d_h = d_h[4:-4].reshape([l,8,139])
d_s = Mirror(d_s).reshape(4l,8,139)
# Get music beat index by the librosa toolkit
beats = librosa.beatidx(m)
value,mask = torch.zeros([l,n,139])
value[:,:4,:] = d_h[:,:4,:]
value[:,-4:,:] = d_h[:,-4:,:]
value[:, beats-4:beats+4,:] = d_s
mask[:,:4,:] = 1
mask[:,-4:,:] = 1
mask[:, beats-4:beats+4,:] = 1
def guidance_sample(m,value,mask,s):
    d = torch.rand([l,n,139])
    # There are 1000 diffusion steps.
    for i in reversed(range(0, 1000)):
    if i > 1000*(1-s):
        #  sample d from step t to step t-1
        d = p_sample(d, m, t)
        # The soft-cue diffusion guidance
        value_ = q_sample(value, t - 1)
        d = value_*mask+(1.0 - mask) * d
        # The hard-cue diffusion guidance
        d[:,:4] = value[:,:4]*mask[:,:4]+(1.0 -
    mask[:,:4] )*d[:,:4]
        d[:,-4:] = value[:,-4:] *mask
    [:,-4:]+(1.0-mask[:,-4:])*d[:,-4:]
    else:
        d = p_sample(d, m, t)
        d[:,:4] = value[:,:4]*mask[:,:4]+(1.0-
    mask[:,:4])*d[:,:4]
        d[:,-4:] = value[:,-4:]*mask[:,-4:]+(1.0-
    mask[:,-4:])*d[:,-4:]

    d = d.reshape([ln, 139])
    return d
```

Listing 1. Pseudocode of the Hard/Soft Diffusion Guidance

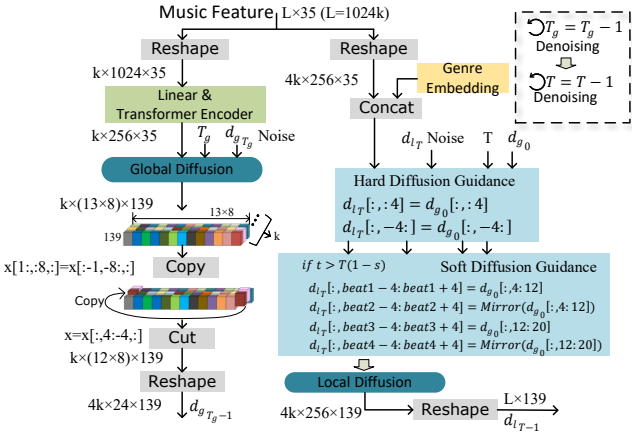

Figure 2. The inference process of Lodge.

C. Details of d_s and d_h

Their primary distinction lies in different purposes. The soft-cue key motion use $\boldsymbol{d_s}$ to guide Local Diffusion to follow the overall choreographic patterns and increase motion

expressiveness. While the primarily purpose of hard-cue key motion $\boldsymbol{d_h}$ is to support parallel generation. Both $\boldsymbol{d_s}$ and $\boldsymbol{d_h}$ are 8-frame key motions generated by Global Diffusion. $\boldsymbol{d_h}$ operates at the beginning and end of Local Diffusion, employing hard diffusion guidance to ensure strict consistency with the initial and final frames of the generated motion, thereby supporting parallel generation. Meanwhile, $\boldsymbol{d_s}$ operates in the middle of Local Diffusion, serving as a soft cue to improve the dance quality.

D. Additional Ablation Studies (tested on the FineDance dataset)

D.1. The Characteristic Dance Primitives

To reduce the computational load of Global Diffusion and to convey global choreography patterns effectively, we propose the Characteristic Dance Primitives. These primitives are dimensionalized as $(l', 8, 139)$, where l' represents the number of dance primitives, '8' denotes the temporal dimension encompassing a continuous sequence of eight frames, and '139' corresponds to the dimensions of the motion feature. However, it is feasible to configure Dance Primitives as discrete frames. Therefore, we conducted a four-fold temporal downsampling of the ground truth dance, which is utilized to train the Global Diffusion for generating discrete dance primitives. To evaluate the relative efficacy of these methodologies, we conduct ablation experiments on the dance primitives as Table 1.

Method	$\text{FID}_k \downarrow$	$\text{Div}_k \uparrow$	BAS $\uparrow$
Ground Truth	/	9.73	0.2120
Discrete	55.17	5.44	0.1969
Continuous	45.56	6.75	0.2397

Table 1. Ablation study of the characteristic dance primitives. 'Discrete' means the dance is generated by the guidance of discrete dance primitives, 'Continuous' means the dance is generated by the guidance of continuous dance primitives

The generated motion guided by discrete dance primitives often results in incoherence, primarily due to the lack of velocity information. This issue is reflected in the increased values of the FID_k[2, 4] as shown in Table 1. Furthermore, the guidance provided by these discrete dance primitives disrupts the beat consistency between music and dance, which consequently leads to a significant decline in the Beat Alignment Score (BAS)[2].

D.2. Ablation Studies of the Hyper-parameter N and n

As described in Section 3.2 of the main paper, N represents the temporal receptive field of the Global Diffusion. The length of global music feature input into Global Diffusion is N. Meanwhile, n denotes the frame number of dance generated by the Local Diffusion.

In this part, we investigate the impact of different N and n. Thanks to our parallel architecture, Lodge can directly generate dance with ln frames, where l is a positive integer. The primary objective of these ablation experiments is to explore how different values affect dance performance.

N	n	$\text{FID}_k \downarrow$	$\text{Div}_k \uparrow$	BAS $\uparrow$
1024	512	61.66	**8.14**	0.1864
1024	256	**45.56**	6.75	**0.2397**
1024	128	45.86	5.54	0.2212
512	256	59.72	5.30	0.2182
512	128	46.74	5.76	0.2124

Table 2. Ablation study of the hyper-parameter N and n.

As shown in Table 2, when n is 512, the quality of motion, as measured by FID_k, deteriorates significantly due to the network's limited capability in modeling long sequences. This also results in a substantial increase in the cost of training Local Diffusion. Comparing cases where n is 128 and 256, we observe only a marginal difference in FID_k. However, crucially, we find that maintaining coherence at this value requires frequent incorporation of d_h within the Hard Diffusion Guidance. Such regular intervention tends to disrupt the overall dance structure. Therefore, we ultimately set n as 256.

Comparing the second and fourth rows, it's evident that when N is set to 1024, all metrics show improved performance. Additionally, a larger N enables more comprehensive modeling of the global dependencies between music and dance. Therefore, we ultimately set N as 1024.

E. Visualization Results

We strongly wish you to watch the video in our project page for more details. We conducted comparisons with state-of-the-art dance algorithms, including FACT[2], MNET[1], Bailando[4], and EDGE[5]. Both FACT and MNET are models based on the Transformer and autoregressive architecture. They encounter significant motion freezing issues during long-duration generation. After several seconds, their motion tends to freeze. Bailando is a model designed based on VQ-VAE[6] and GPT[3]. Its primary limitation lies in the encoding capacity of VQ-VAE, which restricts the network's ability to produce complex dance movements. EDGE is a model based on Diffusion and serves as the backbone of this study. Its main issue is the lack of learning global choreography patterns, resulting in noticeable incoherence at the joints and a relative monotony in the move-

Figure 3. Compare with the SOTAs.

ments. Our method, benefiting from the Coarse-to-Fine architecture, along with the Characteristic Dance Primitives and the Foot Refine Block, is capable of generating coherent, high-quality, and expressive dance sequences.

References

[1] Jinwoo Kim, Heeseok Oh, Seongjean Kim, Hoseok Tong, and Sanghoon Lee. A brand new dance partner: Music-conditioned pluralistic dancing controlled by multiple dance genres. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3490–3500, 2022. 2

[2] 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] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018. 2

[4] 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

[5] 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

[6] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017. 2