# Supplementary Material of EgoMusic-driven Human Dance Motion Estimation with Skeleton Mamba

Quang Nguyen[1], Nhat Le[2], Baoru Huang[7], Minh Nhat Vu[3], Chengcheng Tang[4], Van Nguyen[1], Ngan Le[5], Thieu Vo[6], Anh Nguyen[7]

[1]FPT Software AI Center [2]The University of Western Australia [3]TU Wien [4]Meta

[5]University of Arkansas [6]National University of Singapore [7]University of Liverpool * Corresponding author

https://zquang2202.github.io/SkeletonMamba/

## SUMMARY

This Supplementary Material provides extra material for the paper "*EgoMusic-driven Human Dance Motion Estimation with Skeleton Mamba*". The material is organized as follows:

- Section 1 provides mathematical background for State Space Models and Diffusion Model.
- Section 2 provides mathematical proof for our theory in the main paper.
- Section 3 discusses the motivation and practical impact of EgoMusic-driven Human Dance Estimation.
- Section 4 provides more details on our Human Tokenizer.
- Section 5 provides more examples and analysis of our EgoAIST++ dataset.
- Section 6 outlines the detailed implementation of our method and baseline setup.
- Section 7 presents the proposed MMV metrics and additional experiments, including a user study, analysis of failure cases, extended visualizations of cross-dataset experiments, and inference time.
- Section 8 demonstrates the detail implementation of our Skeleton Mamba when we apply it to the text-to-motion generation and human action recognition tasks.
- Section 9 discusses some interesting future directions.

## 1. Preliminary

**State space models (SSM).** SSM aims to transform an input sequence $x(t) \in \mathbb{R}$ to the output sequence $y(t) \in \mathbb{R}$ using the following equation:

$$h'(t) = \hat{\mathbf{A}}h(t) + \hat{\mathbf{B}}x(t), \quad y(t) = \hat{\mathbf{C}}h(t), \tag{1}$$

where $h(t) \in \mathbb{R}^{N\times 1}$ represents the hidden state, and $\hat{\mathbf{A}} \in \mathbb{R}^{N\times N}$, $\hat{\mathbf{B}} \in \mathbb{R}^{N\times 1}$, and $\hat{\mathbf{C}} \in \mathbb{R}^{1\times N}$ serve as projection matrices, with $N$ denoting the number of states. To adapt this continuous-time formulation for deep learning applications with discrete data, consider a multivariate input sequences $\mathbf{x} = [x_1, \dots, x_T] \in \mathbb{R}^{L\times D}$ with $\forall t, x_t \in \mathbb{R}^D$, Mamba [8] first generates parameters as $\hat{\mathbf{B}}, \hat{\mathbf{C}} = \mathbf{W}_B\mathbf{x}, \mathbf{W}_C\mathbf{x} \in \mathbb{R}^{D\times 1}$, $\hat{\mathbf{A}} = \mathbf{X}\mathbf{W}_A \in \mathbb{R}^L$, where $\mathbf{W}_B, \mathbf{W}_C \in \mathbb{R}^{D\times N}$, $\mathbf{W}_A \in \mathbb{R}^{D\times 1}$ are learnable matrices.

To obtain a discrete-time variant of SSM, a zero-order hold discretization is applied, leading to the following formulation:

$$\begin{aligned} h_t &= \mathbf{A}_t h_{t-1} + \mathbf{B}_t^\top x_t, \\ y_t &= \mathbf{C}_t h_t, \end{aligned} \tag{2}$$

where $\mathbf{A}_t = e^{\Delta A_t} \in \mathbb{R}^{N\times N}$, $\mathbf{B}_t = (\Delta A_t)^{-1}(e^{\Delta A_t} - I) \cdot \Delta\hat{\mathbf{B}}$, $\mathbf{C}_t = \hat{\mathbf{C}}$, $y \in \mathbb{R}^{L\times D}$, $h \in \mathbb{R}^{N\times D}$. In this formulation, $\hat{\mathbf{A}} \in \mathbb{R}^{N\times N}$ is a learnable diagonal matrix, and all projection matrices $\mathbf{A}_t, \mathbf{B}_t, \mathbf{C}_t$ enable the linear time-variant discrete system that selectively attends to inputs $x$ and hidden state $h$ of each timestamp $t$.

**State Space Duality (SSD).** The Structured State Space Duality (SSD) model [4] builds upon the State Space Model (SSM) [8], providing significant improvements in computational efficiency, particularly in terms of speed and memory usage. Instead of using a full diagonal evolution matrix $\hat{\mathbf{A}}$, SSD simplifies it into a scalar form $\hat{a} \in \mathbb{R}$, leading to $\mathbf{a} \in \mathbb{R}^L$ through an equivalent discretization process. With this simplification, SSD reformulates Equation (2) into a matrix transformation:

$$y = \mathrm{SSD}(\mathbf{x}, \mathbf{a}, \mathbf{B}, \mathbf{C}) = (\mathbf{M} \odot (\mathbf{C}\mathbf{B}^\top))\mathbf{x}, \tag{3}$$

$$\mathbf{M}_{ij} = \begin{cases} \prod_{k=j+1}^{i} a_k & \text{if } i > j, \\ 1 & \text{if } i = j, \\ 0 & \text{if } i < j, \end{cases} \tag{4}$$

where $\mathbf{M} \in \mathbb{R}^{L\times L}$ represents a transformation matrix, and

$\odot$ denotes the Hadamard product. This reformulation improves efficiency while maintaining the expressive power of the original model.

**Diffusion Model.** Denosing Diffusion Probabilistic Models (DDPMs) [11] are a class of generative models that learn to gradually remove noise from a noisy input $\mathbf{x}_t$ over a series of steps, where each step has a different noise level $t$. The noise accumulation is described by

$$q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\sqrt{\alpha_t}\mathbf{x}_0, (1-\alpha_t)\mathbf{I}) \ , \tag{5}$$

where $\mathbf{x}_0$ represents the original clean data, $\alpha_t = \prod_{s=1}^{t}(1-\beta_s)$, and $\beta_t$ controls the noise schedule. The objective of the conditional diffusion model is to learn the condition distribution $p_\theta(\mathbf{x}_0|\mathbf{C})$. As described in [6, 11], the denoising model $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t), \mathbf{C}$, parameterized by $\theta$, is trained to reverse this process by estimating the Gaussian posterior $q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)$.

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \mu_\theta(\mathbf{x}_t, t), \mathbf{\Sigma}_\theta(\mathbf{x}_t, t)) \ , \tag{6}$$

$$\mu(\mathbf{x}_t, t) = \frac{1}{\sqrt{\alpha_t}}(\mathbf{x} - \frac{\beta_t}{\sqrt{1-\hat{\alpha}_t}}\epsilon_\theta(\mathbf{x_t}, \mathbf{t})) \ , \tag{7}$$

where $\hat{\alpha}_t = \sum_{s=1}^{t}\alpha_s$, and $\mathbf{\Sigma}_t$ is a variance scheduler of choice. The training loss is typically defined as a reconstruction loss for the mean or the clean input $x_0$, which takes the following simplified form:

$$\begin{aligned}\mathcal{L}_{\text{simple}}(\theta) &= \mathbb{E}_{t,\epsilon,\mathbf{x}_0}\left[\|\epsilon - \epsilon_\theta\left(\sqrt{\hat{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\hat{\alpha}_t}\epsilon, t\right)\|^2\right] \\ &= \mathbb{E}_{t,\epsilon,\mathbf{x}_0}\left[\|\epsilon - \epsilon_\theta(\mathbf{x}_t, t)\|^2\right] \ .\end{aligned} \tag{8}$$

**Condition Diffusion Model.** The objective of the conditional diffusion model is to learn the condition distribution $p_\theta(\mathbf{x}_0|\mathbf{C})$. To achieve this, we modify the diffusion framework by including the condition $\mathbf{C}$ as part of the input for the reverse process, represented as:

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{C}) = \mathcal{N}(\mathbf{x}_{t-1}; \mu_\theta(\mathbf{x}_t, t, \mathbf{C}), \Sigma_\theta(\mathbf{x}_t, t, \mathbf{C})) \ , \tag{9}$$

$$\begin{aligned}\mathcal{L}_{\text{simple}}(\theta) &= \mathbb{E}_{t,\epsilon,\mathbf{x}_0}\left[\|\epsilon - \epsilon_\theta\left(\sqrt{\hat{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\hat{\alpha}_t}\epsilon, t, \mathbf{C}\right)\|^2\right] \\ &= \mathbb{E}_{t,\epsilon,\mathbf{x}_0}\left[\|\epsilon - \epsilon_\theta(\mathbf{x}_t, t, \mathbf{C})\|^2\right] \ .\end{aligned} \tag{10}$$

## 2. Proof of Theorem 1

*Proof.* The Skeleton Mamba architecture can be simplified into three primary components: the Human Tokenizer (HT), the Inverse Human Tokenizer ($\text{HT}^{-1}$), the Group Scan (GS), and the Joint Scan (JS). This discussion focuses on the spatial processing capabilities of the Skeleton Mamba architecture, excluding the Temporal Scan, as its primary role is to handle temporal dependencies. The function $f(\cdot)$, derived from the Skeleton Mamba architecture, processes a frame of motion $\mathbf{x} \in \mathbb{R}^{J\times D}$ as follows:

$$f(\mathbf{x}) = \text{HT}^{-1}(\text{JS}(\text{GS}(\text{HT}(\mathbf{x})))) \ . \tag{11}$$

The function first tokenizes the human pose $\mathbf{x} \in \mathbb{R}^{J\times D}$ into $G$ group tokens:

$$\mathbf{g} = [\mathbf{g}_1, \mathbf{g}_2, \ldots, \mathbf{g}_G] = \text{HT}(\mathbf{x}) \ , \tag{12}$$

where $\mathbf{g}_i \in \mathbb{R}^{G\times E}$ with $E = P \times D$. Then the Group Scan is applied to the sequence $\mathbf{g}$ to achieve transformed sequence $\mathbf{y}$:

$$\begin{aligned}\mathbf{y} = [\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_G] &= \text{GS}(\mathbf{g}) \\ &= \text{Mean}([\text{SSD}(\text{Concat}(\mathbf{g}^{\pi_1}, \ldots, \mathbf{g}^{\pi_n}))])\end{aligned} \tag{13}$$

Given that each token at the $i$-th position can access previous tokens through the State space model mechanism, we split the SSD into $n$ smaller modules, denoted as $\widetilde{\text{SSD}}$:

$$\text{GS}(\mathbf{g}) = \frac{1}{n}\sum_{i=1}^{n}\left[\widetilde{\text{SSD}}_i(\mathbf{g}^{\pi_1}, \ldots, \mathbf{g}^{\pi_i})\right]^{\pi_i^{-1}} \ . \tag{14}$$

Each group embedding $\mathbf{y}_i$ to a sequence of individual joints represented as $\mathbf{y}'_i \in \mathbb{R}^{T\times P\times D}$ using a linear layer and rearrange operator. In Joint Scan, $n$ SSD modules with shared parameters are applied, where each module processes a token $\mathbf{y}'_i$ independently:

$$\mathbf{y}''_i = \text{SSD}_i(\mathbf{y}_i) \ . \tag{15}$$

All outputs of the SSD modules are then concatenated and fed to the Inverse Human Tokenizer to restore the original pose shape $\mathbf{t} \in \mathbb{R}^{T\times J\times D}$.

$$\mathbf{t} = \text{HT}^{-1}((\text{Concat}(\mathbf{y}''_1, \mathbf{y}''_2, \ldots, \mathbf{y}''_G))) \ . \tag{16}$$

Let's analyze the permutation equivariant property of $\text{GS}(\cdot)$. With $\rho \in Sym(G)$, we have:

$$\begin{aligned}\text{GS}(\mathbf{g}^\rho) &= \frac{1}{n}\sum_{i=1}^{n}\left[\widetilde{\text{SSD}}_i(\mathbf{g}^{\rho\pi_1}, \ldots, \mathbf{g}^{\rho\pi_i})\right]^{\pi_i^{-1}} \\ &= \frac{1}{n}\left\{\sum_{i=1}^{n}\left[\widetilde{\text{SSD}}_i(\mathbf{g}^{\rho\pi_1}, \ldots, \mathbf{g}^{\rho\pi_i})\right]^{(\rho\pi_i)^{-1}}\right\}^{\rho} \\ &\neq \text{GS}(\mathbf{g})^\rho \ .\end{aligned} \tag{17}$$

It is evident that $\text{GS}(\cdot)$ lacks the permutation equivariant property, which means that the function $f(\cdot)$ also does not possess this property. Thanks to the universal approximation property of the state space model [35], we could approximate the $\text{GS}(\cdot)$ by the function with the permutation equivariant property. We construct a function $\mathcal{F}(\cdot)$ that derived base on $n$ shared weight SSD module:

$$\mathcal{F}(\mathbf{g}) = \frac{1}{n}\sum_{i=1}^{n}[\text{SSD}_i(\mathbf{g}^{\pi_i})]^{\pi_i^{-1}} \ . \tag{18}$$

With $\rho \in Sym(G)$, we also have:

$$\begin{aligned}\mathcal{F}(\mathbf{g}^{\rho}) &= \frac{1}{n}\sum_{i=1}^{n}\left[\mathrm{SSD}_i(\mathbf{g}^{\rho\pi_i})\right]^{\pi_i^{-1}} \\ &= \frac{1}{n}\sum_{i=1}^{n}\left[\mathrm{SSD}_i(\mathbf{g}^{\rho\pi_i})\right]^{(\rho\pi_i)^{-1}\rho}\ .\end{aligned} \tag{19}$$

Assume that $\{\rho\pi_1, \dots, \rho\pi_n\} = \{\pi_1, \dots, \pi_n\}$, then we have:

$$\begin{aligned}\mathcal{F}(\mathbf{g}^{\rho}) &= \frac{1}{n}\sum_{i=1}^{n}\left[\mathrm{SSD}_i(\mathbf{g}^{\rho\pi_i})\right]^{\pi_i^{-1}} \\ &= \left[\mathcal{F}(\mathbf{g})\right]^{\rho}\ .\end{aligned} \tag{20}$$

This indicates that the function $\mathcal{F}(\cdot)$ is an $H$-equivariant function. According to [35], the State Space Model is a universal approximator. Therefore, it follows from [21] that the function $\mathcal{F}(\cdot)$, which is a frame averaging, is a universal approximator of $H$-equivariant functions from $\mathbb{R}^{G\times E}$ to $\mathbb{R}^{G\times E}$.

To prove the universality of our model, we will choose $\mathrm{JS}(\cdot)$ to be the identity mapping. It is noted that, if our model with this fixed $\mathrm{JS}(\cdot)$ is a universal approximator, then so is our model with parameterized $\mathrm{JS}(\cdot)$. Since $\mathcal{F}(\cdot)$ is a universal approximator of $H$-equivariant functions from $\mathbb{R}^{G\times E}$ to $\mathbb{R}^{G\times E}$, and $\mathrm{HT}(g(\cdot))$ is an $H$-equivariant functions from $\mathbb{R}^{G\times E}$ to $\mathbb{R}^{G\times E}$. We can choose a function $\mathcal{F}(\cdot)$ satisfies:

$$||\mathrm{HT}(g(\mathbf{g})) - \mathcal{F}(\mathbf{g})||_\infty < \frac{\epsilon}{2},\ \forall \mathbf{g} \in \mathrm{HT}(\mathbf{x}), \tag{21}$$

or consequently:

$$||g(\mathbf{x}) - \mathrm{HT}^{-1}(\mathrm{JS}(\mathcal{F}(\mathrm{HT}(\mathbf{x}))))||_\infty < \frac{\epsilon}{2}, \tag{22}$$

for all $\mathbf{x} \in K \subseteq \mathbb{R}^{J\times D}$. Moreover, the Group Scan, which is basically an SSM, is also a universal approximator [35]. There is a Group Scan function such that:

$$||\mathrm{GS}(\mathbf{g}) - \mathcal{F}(\mathbf{g})||_\infty < \frac{\epsilon}{2}\ , \tag{23}$$

for all $\mathbf{g} \in \mathrm{HT}(\mathbf{x})$, or consequently:

$$\begin{aligned}&||f(\mathbf{x}) - \mathrm{HT}^{-1}(JS(\mathcal{F}(\mathrm{HT}(\mathbf{x}))))||_\infty \\ &= ||\mathrm{HT}^{-1}(\mathrm{JS}(\mathrm{GS}(\mathrm{HT}(\mathbf{x})))) - \mathrm{HT}^{-1}(\mathrm{JS}(\mathcal{F}(\mathrm{HT}(\mathbf{x}))))||_\infty \\ &< \frac{\epsilon}{2},\end{aligned} \tag{24}$$

for all $\mathbf{x} \in K \subseteq \mathbb{R}^{J\times D}$. From Equation 22 and Equation 24, we have:

$$\begin{aligned}||f(\mathbf{x}) - g(\mathbf{x})||_\infty &\le ||f(\mathbf{x}) - \mathrm{HT}^{-1}(\mathrm{JS}(\mathcal{F}(\mathrm{HT}(\mathbf{x}))))||_\infty \\ &\quad + ||\mathrm{HT}^{-1}(\mathrm{JS}(\mathcal{F}(\mathrm{HT}(\mathbf{x})))) - g(\mathbf{x})||_\infty \\ &< \frac{\epsilon}{2} + \frac{\epsilon}{2} \\ &= \epsilon,\ \forall \mathbf{x} \in K \subseteq \mathbb{R}^{J\times D}\ .\end{aligned} \tag{25}$$

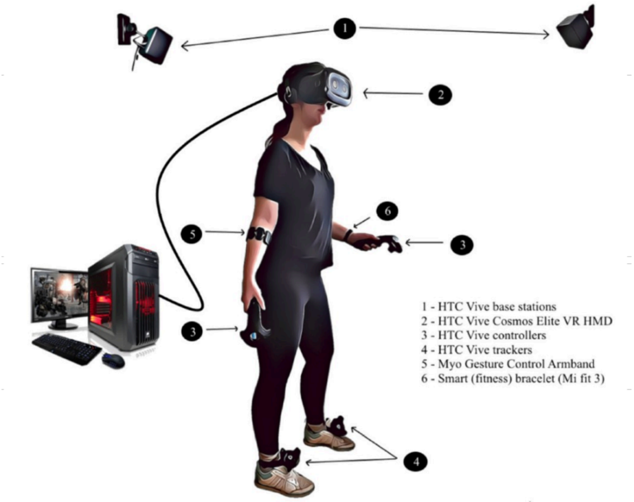
1
2
6
5
3
3
4
1 - HTC Vive base stations
2 - HTC Vive Cosmos Elite VR HMD
3 - HTC Vive controllers
4 - HTC Vive trackers
5 - Myo Gesture Control Armband
6 - Smart (fitness) bracelet (Mi fit 3)

Figure 1. A virtual reality setup with HTC Vive system [29].

Figure 2. An application of ego-music dance estimation.

Thus, the theorem is proven, demonstrating that the Skeleton Mamba architecture can represent complex human motions. This includes motions requiring precise coordination to align with both egocentric views and musical input while preserving the equivariant properties inherent to human body symmetries. □

## 3. Task Motivation

Dance is a fundamental form of human expression, often driven by music and influenced by visual perception. Estimating human motion from egocentric video and music presents a novel and challenging research direction with significant implications for both academic and real-world applications. Current dance motion estimation method relies on third-person cameras [1, 12, 23], which often suffer from occlusion, viewpoint changes, and depth ambiguity. This drawback makes it challenging to capture complex dance movements accurately. To improve the motion estimation accuracy, AR/VR systems incorporate external sensors, such as HTC Vice base stations, controllers, and motion trackers [29], as illustrated in Fig. 1. However, this setup is expensive and less practical for widespread real-world applications. As illustrated in Fig 2, VR dance games demonstrate the potential of immersive experiences driven by body movement, yet they still depend on external tracking systems. To overcome these challenges, a promising approach is to estimate dance poses directly from egocentric video and music, eliminating the need for external hardware while ensuring a more accessible and practical solution.

## 4. Human Tokenizer Details

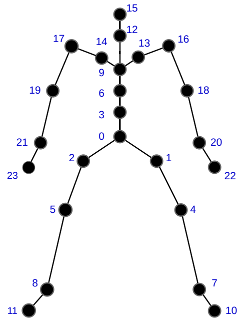

Figure 3. **Joint index of the human body.**

We provide the joint indices of the human body in Fig. 3 for clarity and ease of reference. Let $S_J = Sym(J)$ be the symmetry group of $J$ elements and $\sigma \in S_J$. The Human Tokenizer (HT) and Inverse Human Tokenizer are designed such that they can reflect the human body's symmetries. In other words, the Human Tokenizer and Inverse Human Tokenizer are chosen such as the set $H = \{\sigma \in S_J | \text{HT}(\mathbf{x}^\sigma) = \text{HT}(\mathbf{x})\}$ is a non-empty subset of $S_J$. For example, the human body can be divided into five distinct groups, with each group containing six joints. The joint indices for these groups are as follows: *i)* Spine and head: (0, 3, 6, 9, 12, 15); *ii)* Right arm: (9, 14, 17, 19, 21, 23); *iii)* Left arm: (9, 13, 16, 18, 20, 22); *iv)* Right leg: (0, 2, 3, 5, 8, 11), *v)* Left leg: (0, 1, 3, 4, 7, 10). This grouping captures the inherent symmetries of the human body. For instance, groups representing the left and right limbs (e.g., arms or legs) can be interchanged without affecting the overall structural representation. Such symmetrical grouping effectively models the bilateral relationships and similar movement patterns between corresponding parts on either side of the body.

## 5. EgoAIST++ Dataset Visualization

Our EgoAIST++ dataset combines dance sequences from the AIST++ dataset [17] with 3D mesh scenes from the Replica dataset [30]. To construct the dataset, we divide the motion sequences into multiple 5-second subsequences. Each subsequence is then positioned within the 3D mesh scene at a randomly selected location and orientation. We ensure that the dancer's feet maintain contact with the floor of the 3D scene. To validate the placement, we calculate a penetration score $S_p$ for each human mesh in the subsequence relative to the 3D scene, using a method by [34]. If the penetration score exceeds a predefined threshold ($S_p >$ threshold), indicating potential collisions between the motion and objects in the scene, the current sequence is discarded, and a new random placement is attempted. In practice, we set the threshold to threshold $= 2$. All the valid motions are manually verified before extracting the corresponding camera translation and rotation. The camera is positioned at the mesh index located between the eyes, simulating a forward-facing egocentric perspective. In this setup, most of the human body is not visible in the egocentric images, encouraging the model to infer motion primarily from changes in the surrounding environment, even in the absence of explicit visual cues from the human body. We illustrate examples of valid placements in Fig. 4. We also provide more egocentric examples from our EgoAIST++ dataset in Fig. 5.

## 6. Implementation Details

### 6.1. Auxiliary loss

We incorporate auxiliary losses (Section 4.4 in our Main Paper) to enhance the physical realism of the generated motion. Specifically, we employ position loss and velocity loss, similar to those introduced in [13, 27, 32]. The position loss $\mathcal{L}_{pos}$ and the velocity loss $\mathcal{L}_{vel}$ are defined as:

$$\mathcal{L}_{pos} = \frac{1}{T}\sum_{i=1}^{T} \|\text{FK}(\mathbf{x}^i) - \text{FK}(\hat{\mathbf{x}}^i)\|_2^2 \,, \tag{26}$$

$$\mathcal{L}_{vel} = \frac{1}{T-1}\sum_{i=1}^{T-1} \|(\mathbf{x}^{i+1} - \mathbf{x}^i) - (\hat{\mathbf{x}}^{i+1} - \hat{\mathbf{x}}^i)\|_2^2 \,, \tag{27}$$

where $\text{FK}(\cdot)$ denotes the forward kinematics function that converts rotation angle to position, $\mathbf{x}^i$ is the ground truth pose, and $\hat{\mathbf{x}}^i$ is the predicted pose at frame $i$. The contact loss $\mathcal{L}_{contact}$ is applied to minimize foot sliding:

$$\mathcal{L}_{contact} = \frac{1}{T-1}\sum_{i=1}^{T-1} \|(\text{FK}(\hat{\mathbf{x}}^{i+1}) - \text{FK}(\hat{\mathbf{x}}^i)) \cdot \hat{b}^i\|_2^2 \,, \tag{28}$$

where $\hat{b}^{(i)}$ is a binary indicator for ground contact. The kinematic loss is expressed as follows:

$$\mathcal{L}_{kin} = \lambda_{pos}\mathcal{L}_{pos} + \lambda_{vel}\mathcal{L}_{vel} + \lambda_{contact}\mathcal{L}_{contact} \,, \tag{29}$$

where $\lambda_{pos}$, $\lambda_{vel}$, and $\lambda_{contact}$ are hyperparameter controlling the weight of the corresponding losses.

### 6.2. EMM network details

For a sequence of egocentric images, denoted as $\mathbf{v} \in \mathbb{R}^{T \times C \times H \times W}$, we utilize a ResNet [10] followed by an MLP

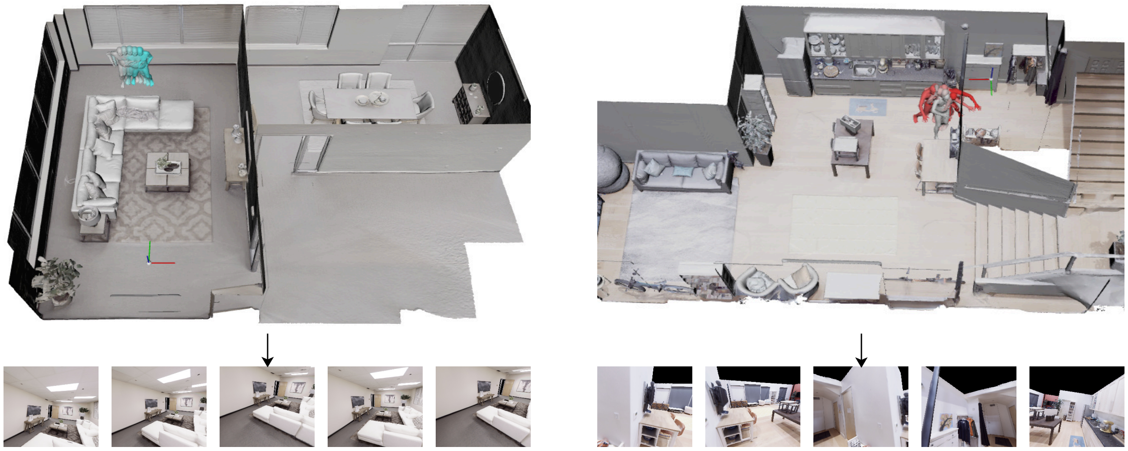

Figure 4. **EgoAIST++ dataset visualization.** Top row shows the human motion within 3D scene. Second row shows the egocentric view.

| Components | Description | Input size | Output size |
|---|---|---|---|
| (1) Audio Encoder | A backbone extracting music features comprised of Jukebox and two layers of Transformer | $[T \times L]$ | $[T \times D_c]$ |
| (2) Vision Encoder | A backbone extracting music features comprised of ResNet50 and a MLP layer | $[T \times C \times W \times H\ ]$ | $[T \times D_c]$ |
| (3) Fusion Module | A module to fusion music and vision embedding comprised of two layers of Transformer | $[T \times D_c]$, $[T \times D_c]$ | $[T \times D_c]$ |
| (5) Human Tokenizer | Grouping human skeleton into multiple groups | $[T \times J \times D]$ | $[T \times G \times E]$ |
| (6) Group Scan | Apply MSSD to group sequence | $[T \times G \times E]$ | $[T \times G \times E]$ |
| (7) Joint Scan | Apply n shared weight SSD to joint sequence | $[G \times (T \times 1 \times E)]$ | $[G \times (T \times P \times D)]$ |
| (5) Inverse Human Tokenizer | Reverse the group sequence to original human pose shape | $[T \times G \times P \times D]$ | $[T \times J \times D]$ |
| (8) Temporal Scan | Apply bidirectional SSD to temporal dimension | $[J \times T \times D]$ | $[J \times T \times D]$ |
| (9) Cross Attention | Cross Attention layer to fuse the condition embedding and the pose embedding | $[T \times J \times D]$, $[T \times D_c]$ | $[T \times J \times D]$ |

Table 1. **EgoMusic Motion Network (EMM) Architecture Summarization.**

| Hyperparameter | Value |
|---|---|
| $D$ | 128 |
| $D_c$ | 256 |
| $G$ | 5 |
| $P$ | 6 |
| $J$ | 24 |
| $E$ | 768 |
| $T$ | 150 |
| Motion FPS | 30 |
| Diffusion Step | 1000 |
| Optimizer | AdamW |
| Learning rate | 2e-4 |
| Num. layers (N) | 8 |
| Num. attention head | 4 |
| MLP dim | 1024 |
| Transformer dim | 512 |

Table 2. **Hyperparameter Details.**

layer to extract vision embeddings, represented as $\mathbf{z}_v \in \mathbb{R}^{T \times D_c}$. For the music input $\mathbf{a} \in \mathbb{R}^{T \times L}$, we employ Jukebox [5] along with two Transformer Encoder layers [33] to generate music embeddings, denoted as $\mathbf{z}_a \in \mathbb{R}^{T \times D_c}$. The fusion of these aligned vision and music embeddings is performed using our Fusion Module, consisting of two Transformer Encoder layers [33]. The architecture and hyperparameters of our model, EMM, are summarized in Table 1 and Table 2.

### 6.3. Baseline details

**Vision-conditioned versions of FACT [17], Bailando [27], and EDGE [32].** We slightly modify these baselines to fuse egocentric images and music features as input. For a fair comparison, as in our backbone, we use the same backbone [10] to extract features from the egocentric images. We employ our Fusion Module to combine the egocentric and music features. In addition to the training losses and parameters from the original models, we also incorporate an alignment loss as proposed in our method to improve the consistency between the modalities. At the sampling stage, we also apply the head guidance goal $\mathcal{G}_{head}(\cdot)$.

**Music-conditioned version of EgoEgo [15].** Similar to our music encoder, we use Jukebox [5] and a transformer encoder [33] to extract music features. We then use our Fusion Module and apply alignment loss to fuse the music features with the head pose features extracted during the first stage to generate a unified condition embedding.

**Kinpoly [20], PoseReg [38].** We employ Jukebox [5] to extract music features. The music feature is then fused with the optical flow feature from their vision encoder. We also employ the alignment loss described in our main paper. All other training losses and parameters are carried over from the original implementation.

## 7. Additional Experiments

### 7.1. Motion-Music-Vision (MMV) metric

To evaluate how well the generated dance motions align with the music and egocentric video, we propose a new Motion-Music-Vision (**MMV**) correlation score:

$$\text{MMV} = \frac{1}{2}\text{MM}(B_x, B_y) + \frac{1}{2}\text{MV}(B_{x'}, B_z)\ ,$$

$$\text{MM}(B_x, B_y) = \frac{1}{|B_y|}\sum \exp\left(-\frac{\min\limits_{\forall t_{x_i} \in B_x} ||t_{x_i} - t_{y_j}||^2}{2\sigma^2}\right)\ ,$$

$$\text{MV}(B_{x'}, B_z) = \frac{1}{|B_z|}\sum \exp\left(-\frac{\min\limits_{\forall t_{x'_i} \in B_{x'}} ||t_{x'_i} - t_{z_k}||^2}{2\sigma^2}\right)\ , \tag{30}$$

where $B_x = \{t_{x_i}\}$ represents the set of kinematic beats extracted from the motion, $B_{x'} = \{t_{x'_i}\}$ represents the head kinematic beats of generated motion, $B_y = \{t_{y_j}\}$ represents the set of music beats, and $B_z = \{t_{z_k}\}$ represents the local minima of optical flow magnitude extracted from egocentric videos. The term $\text{MM}(B_x, B_y)$ corresponds to the music-motion alignment score, as defined in previous works [27]. The term $\text{MV}(B_{x'}, B_z)$ represents the motion-vision alignment score, reflecting how closely the head movements correlate with the egocentric video.

### 7.2. User study

We conducted a user study to evaluate our method and other approaches, 25 participants with different backgrounds and ages from 18-50, were asked to assess the results based on three criteria: the physical plausible of the dancing motions (Physical Plausible), the alignment of body movements with the music (Motion-Music Alignment), and the alignment of head movements with the egocentric video (Motion-Vision Alignment). Participants rated each criterion on a scale from 0 to 5, where 0 indicated very poor, 1 indicated poor, 2 indicated fair, 3 indicated good, 4 indicated very good, and 5 indicated excellent. These scores were then normalized to a range of [0, 1]. Each participant reviewed 20 samples, with 5 samples per method: EgoEgo [15], Edge [32], Ground Truth, and ours. Overall, the results from Fig. 6 show that our method is more favorable compared to other baselines.

### 7.3. Failure Cases

Although our method achieves promising results, it produces incorrect predictions in challenging cases. First, abrupt movements in the egocentric view, such as sharp head jerks or sudden whiplash-like motions, can introduce motion blur. This blur affects the scene understanding, leading to incorrect estimated motion (Fig. 7). Second, when there is a mismatch between the egocentric video and the music, such as when an energetic, fast-paced song accompanies a calm or stationary visual scene, the generated dance motion may fail to align with either the music's rhythm or the visual input's dynamics (Fig. 8).

### 7.4. Cross-dataset visualization

Fig. 9 presents visualization results using EMM trained on our EgoAIST++ dataset, evaluated on EgoExo4D, a real-world motion capture dataset. The predicted motion closely resembles the ground truth images, demonstrating that our method, despite not being trained on real-world images, generalizes effectively to such data.

### 7.5. Inference time

We compare the model parameters and runtime of all methods in Table 3. For diffusion-based models, the reported runtime corresponds to 1,000 sampling steps. Inference time remains a notable challenge for diffusion-based approaches. Our model, built upon Mamba, demonstrates an advantage in inference time compared to transformer-based models. In particular, our EMM has a faster inference time than EgoEgo [15], FACT [17], and EDGE [32], while being slower than Bailando [27].

| Model | PoseReg | Kinpoly | EgoEgo | FACT | Bailando | EDGE | EMM |
|---|---|---|---|---|---|---|---|
| Params (M) | 15 | 22 | 35.1 | 131.4 | 184.5 | 61.2 | 47.4 |
| Inference Time (s) | 13.2 | 4.5 | 47.3 | 26.4 | 14.1 | 30.1 | 24.8 |

Table 3. **Inference time comparison.**

## 8. Skeleton Mamba on Human Motion

### 8.1. Human action recognition

**Setup.** We propose an adaptation of Skeleton Mamba for this task, as illustrated in Fig. 10. The input skeleton sequence is denoted as $\mathbf{x} \in \mathbb{R}^{T \times J \times D}$, where $T$ is the sequence length, $J$ is the number of joints, and $D$ is the embedding dimension. This sequence is processed through $N$ Skeleton Mamba modules. Each Skeleton Mamba module consists of the Human Tokenizer, Group Scan, Joint Scan, Temporal Scan, and Inverse Human Tokenizer, as detailed in the main paper. Next, Global Average Pooling is applied to aggregate both spatial and temporal features from the entire skeleton sequence, producing an output embedding $\mathbf{o} \in \mathbb{R}^{1 \times 1 \times C}$, where $C$ is the number of classes. This embedding is then passed through a linear layer, followed by a softmax function, to generate the final label prediction.

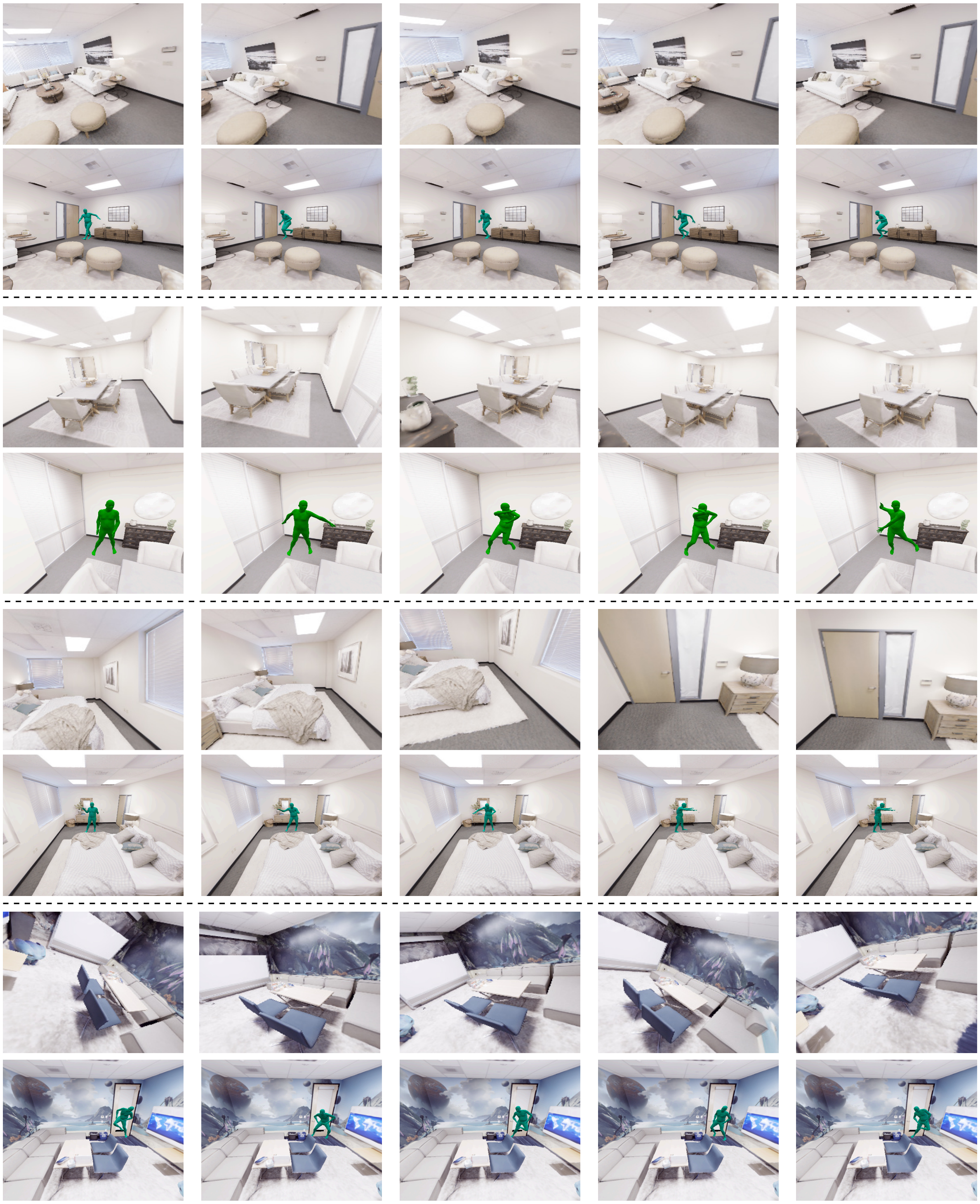

Figure 5. **Samples of our EgoAIST++ dataset.** In each sample, the first row shows the egocentric view from the head-mounted camera, and the second row is the third view.

**Dataset.** We use two datasets: Kinetics400 [14] and NTU RGB+D 60 [18]. The Kinetics 400 dataset is a large-scale dataset with over 260K skeleton sequences from 400 action classes. The NTU RGB+D 120 dataset is a large-scale benchmark for action recognition, comprising nearly 56K skeleton sequences across 60 action classes, captured

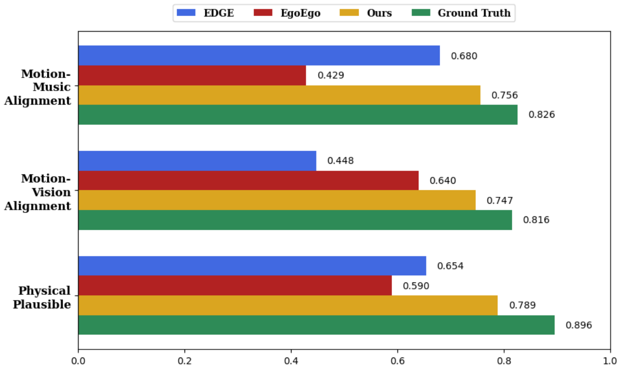

Figure 6. **User study results.**

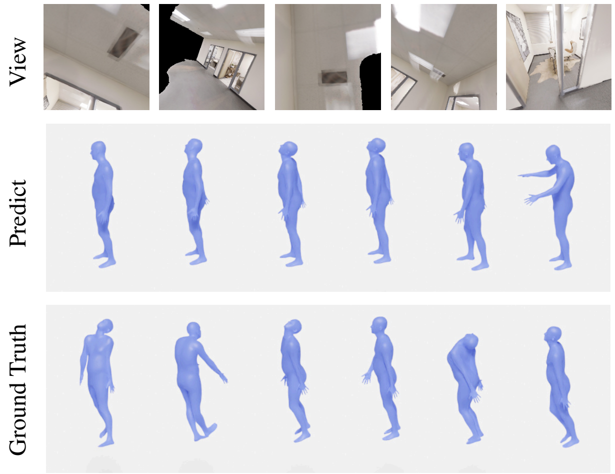

Figure 7. **Failure case 1.** Example of motion blur caused by abrupt head movements, leading to inaccurate motion generation.

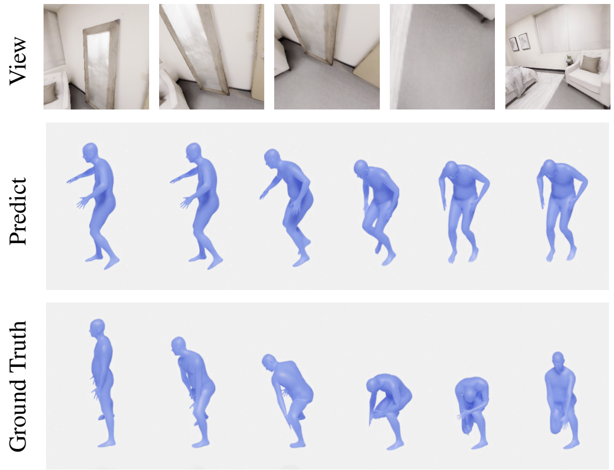

Figure 8. **Failure case 2.** Example of misaligned egocentric view and music, resulting in unsynchronized dance motion.

from 40 unique subjects and 3 distinct camera view angles. The dataset is divided into two evaluation settings: (1) Cross-subject (**X-Sub**), where the training and testing sets consist of different subjects, and (2) Cross-setup (**X-Set**), where the division is based on different camera setups.

| Method | Backbone | NTU60-XSub | NTU60-Xview | Kinetics |
|---|---|---|---|---|
| ST-GCN [36] | GCN | 81.5 | 88.3 | 30.7 |
| AS-GCN [16] | GCN | 86.8 | 94.2 | 34.8 |
| RA-GCN [28] | GCN | 87.3 | 93.6 | - |
| AGCN [25] | GCN | 88.5 | 95.1 | 36.1 |
| DGNN [24] | GCN | 89.9 | 96.1 | 36.9 |
| FGCN [37] | GCN | 90.2 | 96.3 | - |
| Shift-GC [3] | GCN | 90.7 | 96.5 | - |
| MS-G3D [19] | GCN | 91.5 | 96.2 | 38.0 |
| PoseConv3D [7] | CNN | 93.1 | 95.7 | 47.7 |
| DSTA-Net [26] | Transformer | 91.5 | 96.4 | - |
| STTFormer [22] | Transformer | 89.9 | 95.9 | - |
| MotionBERT [40] | Transformer | 87.7 | 94.1 | - |
| MotionBERT [40] (extra data) | Transformer | 93.0 | **97.2** | - |
| Skeleton Mamba (ours) | Mamba | **94.4** | 96.9 | **52.4** |

Table 4. **Skeleton-based action recognition results.**

**Baselines.** We compare our model to state-of-the-art method in skeleton-based action recognition with three types of backbone: Graph-based method (GCN) [3, 16, 19, 24, 25, 28, 36, 37], Convolution neural network (CNN) [7] and Transformer based method [22, 26, 40].

**Results.** We evaluate the top-1 classification accuracy and present the results in Table 4. The findings show that our approach outperforms all previous methods on two out of three benchmarks. Furthermore, without utilizing additional training data, our method achieves superior performance across all benchmarks. Notably, our model represents the first Mamba-based architecture capable of effectively learning the hierarchical spatial and temporal structures of human motion, demonstrating its strength in modeling complex motion patterns.

### 8.2. Text-to-motion generation

**Setup.** For the text-to-motion generation task, we implement the framework shown in Fig. 11, leveraging the core Skeleton Mamba for dance motion estimation. A CLIP Text Encoder processes the text prompt, while a Cross-Attention layer integrates the condition and motion embeddings.

**Benchmark.** We evaluate our method on HumanML3D [9] dataset. We train our model using diffusion and kinematics loss as in [31]. The model is compared with recent text-to-motion methods: MDM [31], MLD [2], and MotionMamba [39]. We employ the same metrics as in [9]. The results are shown in our main paper.

## 9. Future Works

We see several interesting future problems. First, integrating additional modalities such as scene context and human-object interactions would create a more comprehensive understanding of human motion. Second, modeling long dance motions remains a challenge, requiring advancements in temporal modeling to ensure coherence over extended sequences. Third, we could leverage even subtle visual body cues to enhance motion estimation accuracy.

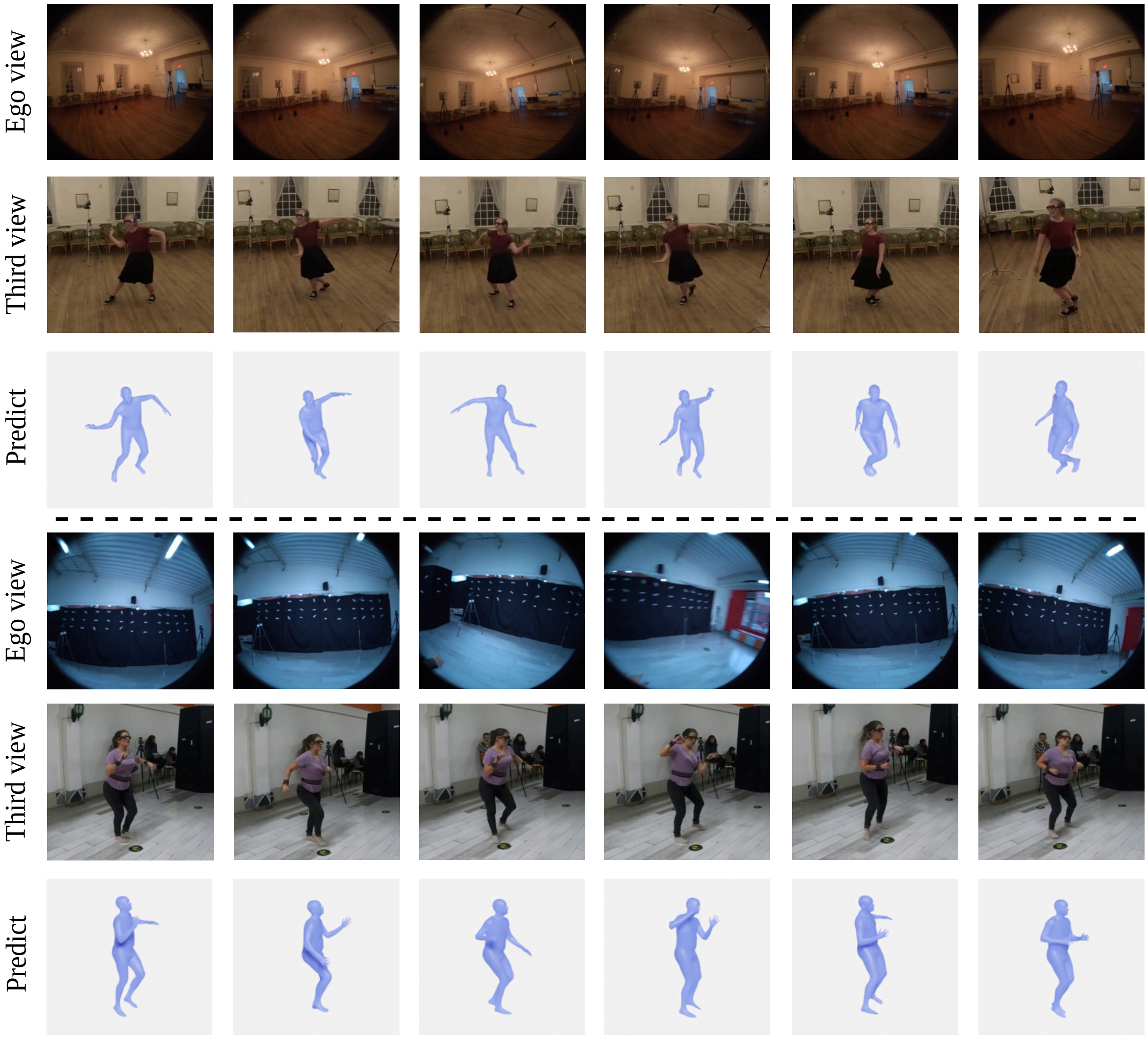

Figure 9. Qualitative results for cross-dataset experiments.

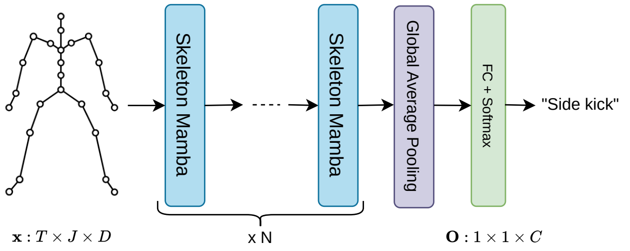

Figure 10. The adaptation of our Skeleton Mamba for skeleton-based human action recognition task.

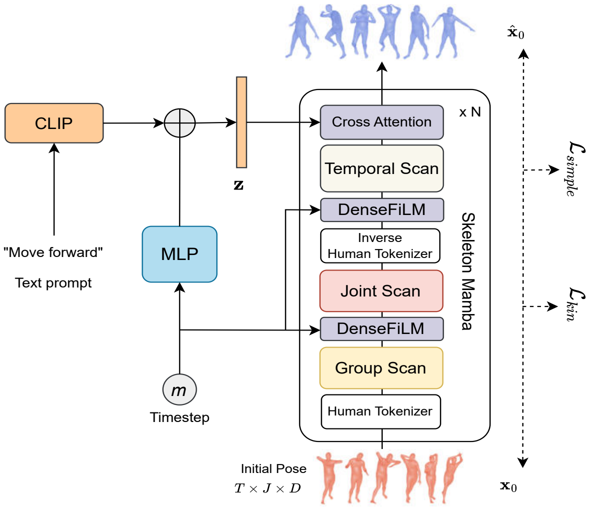

Figure 11. The adaption of our Skeleton Mamba for text-to-human motion task.

## References

[1] Asish Bera, Mita Nasipuri, Ondrej Krejcar, and Debotosh Bhattacharjee. Fine-grained sports, yoga, and dance postures recognition: A benchmark analysis. *TIM*, 2023.

[2] Xin Chen, Biao Jiang, Wen Liu, Zilong Huang, Bin Fu, Tao Chen, and Gang Yu. Executing your commands via motion diffusion in latent space. In *CVPR*, 2023.

[3] Ke Cheng, Yifan Zhang, Xiangyu He, Weihan Chen, Jian Cheng, and Hanqing Lu. Skeleton-based action recognition with shift graph convolutional network. In *CVPR*, 2020.

[4] Tri Dao and Albert Gu. Transformers are SSMs: Generalized models and efficient algorithms through structured state space duality. In *ICML*, 2024.

[5] Prafulla Dhariwal, Heewoo Jun, Christine Payne, Jong Wook Kim, Alec Radford, and Ilya Sutskever. Jukebox: A generative model for music. *arXiv:2005.00341*, 2020.

[6] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *NeurIPS*, 2021.

[7] Haodong Duan, Yue Zhao, Kai Chen, Dahua Lin, and Bo Dai. Revisiting skeleton-based action recognition. In *CVPR*, 2022.

[8] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv:2312.00752*, 2023.

[9] Chuan Guo, Shihao Zou, Xinxin Zuo, Sen Wang, Wei Ji, Xingyu Li, and Li Cheng. Generating diverse and natural 3d human motions from text. In *CVPR*, 2022.

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[11] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *NeurIPS*, 2020.

[12] Xiaodan Hu and Narendra Ahuja. Unsupervised 3d pose estimation for hierarchical dance video recognition. In *ICCV*, 2021.

[13] Zikai Huang, Xuemiao Xu, Cheng Xu, Huaidong Zhang, Chenxi Zheng, Jing Qin, and Shengfeng He. Beat-it: Beat-synchronized multi-condition 3d dance generation. *arXiv:2407.07554*, 2024.

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

[15] Jiaman Li, Karen Liu, and Jiajun Wu. Ego-body pose estimation via ego-head pose estimation. In *CVPR*, 2023.

[16] Maosen Li, Siheng Chen, Xu Chen, Ya Zhang, Yanfeng Wang, and Qi Tian. Actional-structural graph convolutional networks for skeleton-based action recognition. In *CVPR*, 2019.

[17] Ruilong Li, Shan Yang, David A Ross, and Angjoo Kanazawa. Ai choreographer: Music conditioned 3d dance generation with aist++. In *ICCV*, 2021.

[18] Jun Liu, Amir Shahroudy, Mauricio Perez, Gang Wang, Ling-Yu Duan, and Alex C Kot. Ntu rgb+ d 120: A large-scale benchmark for 3d human activity understanding. *TPAMI*, 2019.

[19] Ziyu Liu, Hongwen Zhang, Zhenghao Chen, Zhiyong Wang, and Wanli Ouyang. Disentangling and unifying graph convolutions for skeleton-based action recognition. In *CVPR*, 2020.

[20] Zhengyi Luo, Ryo Hachiuma, Ye Yuan, and Kris Kitani. Dynamics-regulated kinematic policy for egocentric pose estimation. *NeurIPS*, 2021.

[21] Omri Puny, Matan Atzmon, Heli Ben-Hamu, Ishan Misra, Aditya Grover, Edward J Smith, and Yaron Lipman. Frame averaging for invariant and equivariant network design. *ICLR*, 2022.

[22] Helei Qiu, Biao Hou, Bo Ren, and Xiaohua Zhang. Spatio-temporal tuples transformer for skeleton-based action recognition. *arXiv:2201.02849*, 2022.

[23] Challapalli Jhansi Rani and Nagaraju Devarakonda. An effectual classical dance pose estimation and classification system employing convolution neural network–long shortterm memory (cnn-lstm) network for video sequences. *Microprocessors and Microsystems*, 2022.

[24] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. Skeleton-based action recognition with directed graph neural networks. In *CVPR*, 2019.

[25] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. Two-stream adaptive graph convolutional networks for skeleton-based action recognition. In *CVPR*, 2019.

[26] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. Decoupled spatial-temporal attention network for skeleton-based action-gesture recognition. In *ACCV*, 2020.

[27] 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 *CVPR*, 2022.

[28] Yi-Fan Song, Zhang Zhang, Caifeng Shan, and Liang Wang. Richly activated graph convolutional network for robust skeleton-based action recognition. *TSCVT*, 2020.

[29] Iulia-Cristina Stanica, Florica Moldoveanu, Giovanni-Paul Portelli, Maria-Iuliana Dascalu, Alin Moldoveanu, and Mariana Georgiana Ristea. Flexible virtual reality system for neurorehabilitation and quality of life improvement. *Sensors*, 2020.

[30] Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, et al. The replica dataset: A digital replica of indoor spaces. *arXiv:1906.05797*, 2019.

[31] Guy Tevet, Sigal Raab, Brian Gordon, Yoni Shafir, Daniel Cohen-or, and Amit Haim Bermano. Human motion diffusion model. In *ICLR*, 2023.

[32] Jonathan Tseng, Rodrigo Castellon, and Karen Liu. Edge: Editable dance generation from music. In *CVPR*, 2023.

[33] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017.

[34] Jiashun Wang, Huazhe Xu, Jingwei Xu, Sifei Liu, and Xiaolong Wang. Synthesizing long-term 3d human motion and interaction in 3d scenes. In *CVPR*, 2021.

[35] Shida Wang and Beichen Xue. State-space models with layer-wise nonlinearity are universal approximators with exponential decaying memory. *NeurIPS*, 36, 2024.

[36] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *AAAI*, 2018.

[37] Hao Yang, Dan Yan, Li Zhang, Yunda Sun, Dong Li, and Stephen J Maybank. Feedback graph convolutional network for skeleton-based action recognition. *TIP*, 2021.

[38] Ye Yuan and Kris Kitani. Ego-pose estimation and forecasting as real-time pd control. In *ICCV*, 2019.

[39] Zeyu Zhang, Akide Liu, Ian Reid, Richard Hartley, Bohan Zhuang, and Hao Tang. Motion mamba: Efficient and long sequence motion generation. In *ECCV*, 2024.

[40] Wentao Zhu, Xiaoxuan Ma, Zhaoyang Liu, Libin Liu, Wayne Wu, and Yizhou Wang. Motionbert: A unified perspective on learning human motion representations. In *ICCV*, 2023.