AnyMo: Scaling Any-Modality Conditional Motion Generation with Masked Modeling

Yiheng Li[1,2] **Zhuo Li**[3]* **Ruibing Hou**[1]† **Yingjie Chen**[3]
Hong Chang[1,2] **Hao Liu**[3] **Shiguang Shan**[1,2]

[1]Key Laboratory of Intelligent Information Processing,
Institute of Computing Technology, Chinese Academy of Sciences, China
[2]University of Chinese Academy of Sciences, China [3]Independent Researchers
{yiheng.li,zhuo.li}@vipl.ict.ac.cn
{houruibing,changhong,sgshan}@ict.ac.cn
chenyingjie@pku.edu.cn, lewes6369@gmail.com



Figure 1: **Top**: OmniHuMo is a large-scale, high-quality human motion dataset with multimodal annotations. **Bottom**: We present AnyMo, a unified framework for controllable motion generation from diverse modalities and their combinations.

Abstract

Conditional human motion generation remains a fundamental challenge in computer vision and robotics. Despite significant progress, current methods are often constrained by fixed modality configurations and task-specific architectures, leaving cross-modal interactions and the scaling laws of multimodal-conditioned synthesis largely underexplored. A key bottleneck is the scarcity of large-scale modality-aligned motion data, limiting generalization across diverse control signals. In this work, we introduce **OmniHuMo**, a large-scale, high-quality dataset comprising over 5,000 hours of motion and 3.2 million sequences with precisely aligned multimodal annotations (e.g., text, speech, music, and trajectory). Leveraging OmniHuMo, we propose **AnyMo**, a unified multimodal framework combining a Residual FSQ-based motion tokenizer with a scalable masked modeling transformer, enabling high-quality motion synthesis under arbitrary modality combinations. Extensive experiments show that AnyMo achieves high-fidelity synthesis while offering flexible control over both spatial and stylistic attributes. Code and models are available on our project page.

*Project leader.
†Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

Table 1: Comparison with existing motion datasets. "Mono MoCap" refers to markerless monocular video-based motion capture. "Data Agg" denotes datasets constructed by aggregating existing sources.

Datasets	Clips	Duration	Source	Text	Speech	Music	RGB
BEATv2 [27]	1.8K	60H	Marker MoCap	✗	✓	✗	✗
HumanML3D [14]	14.6K	28.6H	Marker MoCap	✓	✗	✗	✗
OpenDanceSet [65]	41K	100.3H	Mono MoCap	✗	✗	✓	✗
Motion-X++ [67]	120K	181H	Data Agg & Mono MoCap	✓	✗	✓	✗
MotionHub [25]	400K	596H	Data Agg	✓	✓	✓	✗
MotionMillion [11]	2M	2000H	Data Agg & Mono MoCap	✓	✗	✗	✗
OmniHuMo(Ours)	3.2M	5048H	Mono MoCap	✓	✓	✓	✓

1 Introduction

Driven by the growing demands of digital media and robotics, human motion generation has advanced rapidly in recent years. The goal is to synthesize realistic and temporally coherent motions under various control signals. Recent progress in generative modeling has enabled motion generation from multiple modalities, including natural language descriptions [64, 13, 4, 36, 49, 58, 42], music [51, 46, 23], speech [6, 27], and spatial trajectories [54, 59, 34].

Despite these advances, achieving precise controllability and robust cross-modal generalization remains challenging due to two key bottlenecks: **First, Scarcity of large-scale, multimodal aligned motion data.** Most existing motion datasets [14, 30, 37, 15] rely on an optical motion capture system, which provides high-fidelity sequences but is costly and labor-intensive, limiting scale and diversity. Recent efforts [67, 65, 11, 55] explore motion extraction from in-the-wild videos, successfully scaling data to a million-level. Nevertheless, these datasets lack comprehensive multimodal alignment (see Tab. 1), reducing their effectiveness for multimodal motion generation. **Second, lack of a multimodally controllable generative framework.** Most existing methods focus on single-modality-driven motion generation [13, 58, 23, 27, 51, 42], limiting flexibility in complex scenarios. Although recent works [20, 6, 68] incorporate multiple modalities within a unified framework, they still follow a "multi-task but single-input" paradigm, treating each modality as an isolated generation task. As a result, they cannot explicitly model cross-modal dependencies required for simultaneous conditioning, and struggle to generalize to arbitrary combinations of control signals for a single motion sequence.

Building on the above challenges, we argue that achieving robust generalization and flexible controllability in human motion generation requires both large-scale multimodal data and scalable model architectures. Accordingly, we identify two key components for any-modality motion generation: 1) a large-scale motion dataset with semantically aligned multimodal annotations, and 2) a scalable framework that supports motion synthesis under arbitrary combinations of input modalities.

To this end, we present **OmniHuMo**, a large-scale human motion dataset with rich multimodal annotations, constructed via an efficient pipeline for automatic labeling of web-scale videos. OmniHuMo offers three key advantages: 1) **Large scale:** over 5,000 hours of motion data (3.2M+ sequences) extracted from web videos, as summarized in Tab. 1; 2) **Multimodal annotations:** all sequences are paired with textual descriptions, with a subset ($\sim$ 500 hours) further annotated with speech or music; 3) **High quality:** rigorous filtering and post-processing ensure reliable motion reconstruction and consistent annotations.

Building upon OmniHuMo, we propose **AnyMo**, a scalable masked modeling framework for motion generation under arbitrary combinations of conditioning signals. The architecture comprises two key components. **1) R-FSQ-based Motion Tokenizer.** While Finite Scalar Quantization (FSQ) [31] improves stability and efficiency over Vector Quantization [52], it still suffers from information loss. To address this, we adopt residual quantization [13], where a base stream captures coarse motion structure and subsequent streams progressively encode residual details, reducing reconstruction error. **2) Scalable Masked Transformer.** We employ a LLaMA-based [50] Transformer backbone for motion token modeling. Unlike autoregressive methods, AnyMo uses bidirectional attention with masked modeling [13] to capture global temporal dependencies across both past and future frames. For multimodal conditioning, modality-specific encoders process heterogeneous inputs, enabling flexible combinations of control signals. In addition, Parallel Mask Modeling is designed to predict

all residual streams simultaneously without sequence flattening, effectively improving efficiency and generation quality.

To the best of our knowledge, OmniHuMo is the largest human motion dataset to date that integrates text, audio, and visual modalities. Extensive experiments demonstrate that AnyMo achieves competitive performance across diverse motion generation tasks, underscoring its superior effectiveness and cross-modal versatility.

2 Related Work

In this section, we review related work on Motion Generation and Generative Masked Transformers. Related work on Data-driven Motion Modeling is provided in Appendix A.1.

Motion Generation with Diverse Modalities. Human motion generation has evolved from single-modality tasks—such as text-driven motion generation [64, 14, 42, 13, 3, 56, 49, 4, 24, 34], audio-conditioned gesture [27, 26, 62, 38] and dance synthesis [51, 46, 23] —toward more advanced paradigms that integrate multiple modalities [20, 6, 68, 32] to enhance controllability. However, due to the scarcity of well-aligned multimodal motion data, most existing methods adopt a shared backbone for multiple single-modality tasks, rather than explicitly modeling cross-modal interactions. Consequently, these approaches struggle to support arbitrary input modalities combinations. This lack of high-quality multimodal alignment remains a key bottleneck for scalable and flexible motion generation.

Generative Masked Transformers. BERT [10] introduces masked modeling for language, where a subset of tokens is randomly masked and a bidirectional Transformer is trained to reconstruct them. This paradigm has been successfully extended to other domains [5, 69, 63]. In motion generation, Generative Masked Modeling (GMM) formulates synthesis as a non-autoregressive "mask-and-in-between" task, offering a favorable trade-off between quality and efficiency. Methods such as MoMask [13] and MMM [36] capture bidirectional temporal dependencies, while BAMM [35] further explores hybrid bidirectional–autoregressive modeling. However, existing GMM approaches are primarily designed for single-modality control and typically model motion as a single-stream token sequence. Their extension to flexible multimodal conditioning and multi-stream token generation remains underexplored. Scaling such frameworks to large datasets remains an open challenge.

3 OmniHuMo Dataset

We introduce **OmniHuMo**, a large-scale **Omni**-modal **Hu**man **Mo**tion dataset, constructed entirely from diverse online videos. To enable scalable and diverse human motion data collection, we design an automated data collection pipeline, as illustrated in Fig. 2. The pipeline consists of five sequential stages: Video Curation; Human 2D Annotation; Human 3D Annotation; Audio Annotation; and Motion Caption Annotation.

3.1 Data Construction Pipeline

Video Curation. Video curation pipeline starts with large-scale video collection from online platforms and public datasets [19]. This process yields over 200 million videos spanning diverse scenarios, actions, and recording conditions. To enhance the robustness of subsequent motion estimation, we perform scene detection and segmentation to mitigate artifacts caused by abrupt temporal transitions. We adopt a coarse-to-fine strategy: PySceneDetect[3] is first used to detect coarse scene boundaries based on pixel intensity and brightness variations, followed by TransNetV2 [47] to refine complex transitions such as fades and wipes. Videos are then segmented into single-scene clips. Finally, we apply strict quality filtering based on heuristics including luminance, bitrate, visual quality, and motion intensity. Details about filtering criteria are provided in Appendix A.2.1.

Human 2D Annotation. To reconstruct human motion in world coordinates, we first extract high-fidelity 2D annotations, including bounding boxes, keypoints, and tracking associations. Specifically,

[3] `https://github.com/Breakthrough/PySceneDetect`

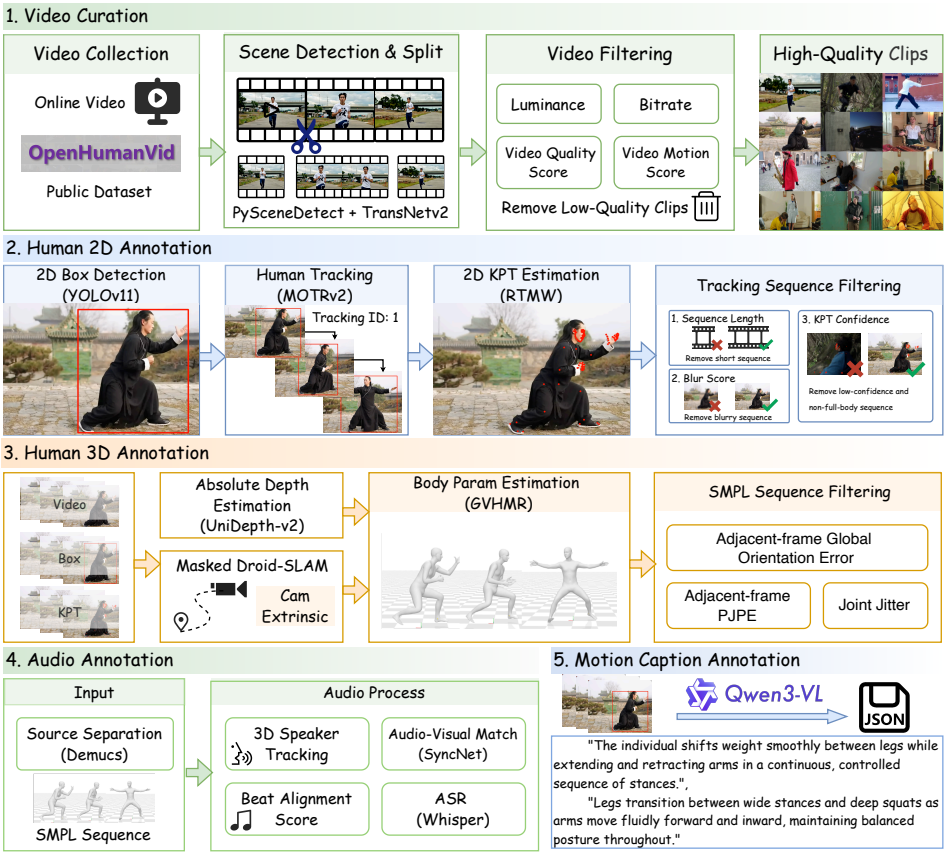


Figure 2: **Data Construction Framework of OmniHuMo.** The proposed pipeline systematically extracts high-quality human motion data with temporally aligned audio signals and corresponding textual descriptions.

YOLOv11 [18] is used for human detection, MOTRv2 [66] for long-term human tracking. Building upon these tracked sequences, 2D human poses are estimated using RTMW [17]. To ensure annotation quality and temporal consistency, we apply three filtering criteria: *1) Temporal duration*: sequences shorter than 60 frames are discarded. *2) Visual fidelity*: sequences with a mean blur score below 0.1 are removed to reduce degradation caused by motion blur. *3) Pose reliability*: frames with an average keypoint confidence below 0.6 are filtered out to reduce severe occlusions and incomplete poses.

Human 3D Annotation. We employ GVHMR [45] to reconstruct 3D human motion in world coordinates. Giving 2D bounding boxes, keypoints, video frames, and camera extrinsics, GVHMR regresses SMPL [28] parameters of 3D human motion sequences. To improve temporal consistency, we further filter out sequences with abrupt root orientation changes or excessive joint jitter caused by camera motion. Detailed criteria is provided in Appendix A.2.2.

Audio Annotation. Audio is an important modality for multimodal motion generation, especially for speech-driven gestures and music-driven dance. We first extract audio tracks from videos and use Demucs [44] to separate vocals and background music. **For dance video identification**, we compute the Beat Alignment Score (BAS) between the music track and the SMPL motion sequence. Samples with BAS above 0.15 are classified as dance sequences, ensuring strong motion-music synchronization. **For speech videos identification**, we adopt a three-stage pipeline: 3D-Speaker [7] is used for speaker tracking, SyncNet [9] for audio-visual synchronization, and Whisper [40] for speech transcriptions. Samples without valid linguistic content are removed.

Motion Caption Annotation. To support semantic understanding and controllable motion synthesis, we generate textual captions for each motion sequence. Source videos are first segmented into clips based on reconstructed motion trajectories. We then use Qwen3-VL-32B [1] to generate fine-grained

action descriptions, with prompts focusing on localized human bounding boxes to capture detailed motion dynamics. Detailed prompt configurations are provided in Appendix A.2.3.

3.2 Dataset Statistics

As summarized in Tab. 1, OmniHuMo contains over **5,000 hours** of human motion data and more than **3.2 million motion sequences**. Due to the heterogeneous video sources, modality annotations are unevenly distributed. Each sequence is paired with 1–3 textual captions, while a subset (approximately 500 hours) additionally contains temporally aligned audio. This design balances large-scale motion diversity with high-quality multimodal annotations, supporting both general motion synthesis and audio-driven generation.

We further analyze motion categories and sequence durations in Fig. 3, Fig. 4 and Fig. 5. OmniHuMo covers diverse activities, from indoor scenarios (e.g., artistic performances and choreographed dance) to outdoor activities (e.g., sports and daily events). Most Sequence are 2–10 seconds long, making them suitable for modeling atomic actions and rapid motion transitions.

4 Method

We propose **AnyMo**, a scalable masked modeling framework for 3D motion generation conditioned on arbitrary combinations of text, music, speech, and trajectory inputs. Trained on the large-scale OmniHuMo dataset, AnyMo generates motion sequences $\mathbf{X} \in \mathbb{R}^{T \times D}$ guided by these multimodal conditions, where T denotes sequence length and D the pose dimensionality.

As illustrated in Fig. 6, AnyMo consists of two key components: 1) a Residual Finite Scalar Quantizer-based tokenizer that discretizes continuous motion into hierarchical tokens (Sec. 4.1); and 2) a scalable masked Transformer built upon LLaMA [50] architecture, which reconstructs motion from masked motion tokens (Sec. 4.2).

4.1 Motion Residual FSQ-VAE

Conventional discrete motion representations typically rely on vector quantization with a single, capacity-limited codebook, which suffers from two key limitations. First, the non-differentiable quantization nearest-neighbor assignment can lead to codebook collapse, leading to highly imbalanced code usage and inefficient utilization of the latent space. Second, single-stage quantization compresses complex motion patterns into a coarse representation, limiting its ability to capture fine-grained dynamics. To address these issues, we adopt **Residual Finite Scalar Quantization (R-FSQ)**, which replaces vector matching by deterministic scalar discretization in a bounded space with hierarchical quantization. This design promotes uniform code utilization and enables progressive multi-scale refinement, leading to high-fidelity reconstruction.

As shown in Fig. 6a, R-FSQ comprises a motion encoder E, decoder D, and $V+1$ hierarchical quantization stages. Given a motion sequence $\mathbf{X} \in \mathbb{R}^{T \times D}$, the encoder E maps it to a continuous latent representation $\mathbf{Z} \in \mathbb{R}^{t \times d}$, where T/t denotes the temporal downsampling ratio. To avoid codebook collapse in conventional VQ, each stage employs a Finite Scalar Quantizer (FSQ). For each vector $\boldsymbol{z} \in \mathbb{R}^d$ within $\mathbf{Z}$, FSQ first applies a bounding function $f(\cdot)$ (instantiated as sigmoid) to

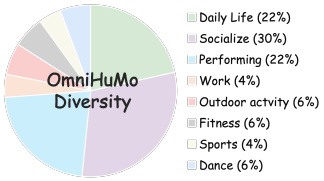


Figure 3: OmniHuMo diversity.

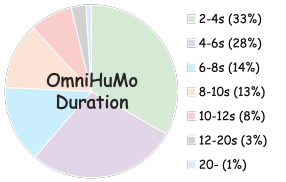


Figure 4: OmniHuMo duration.

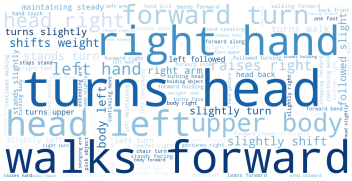
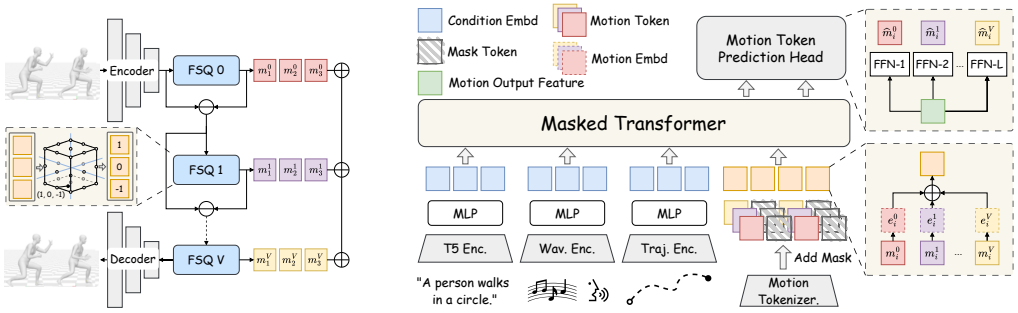


Figure 5: Word cloud of Caption.



(a) Residual FSQ.

(b) Scalable masked transformer architecture.

Figure 6: **Overview of AnyMo**. The framework consists of two components. First, we train a motion tokenizer based on Residual FSQ to discretize continuous motion into multi-stream discrete tokens. Second, we train a masked Transformer that supports diverse conditioning signals, including text, audio, and trajectories, as well as their combinations, to generate coherent human motion sequences.

constrain its value within a predefined range. Each dimension z_i is then discretized into L_i levels:

$$\hat{z}_i = \mathrm{FSQ}\left(z_i\right) = \mathrm{round}\left(f\left(z_i\right) \cdot \left(L_i - 1\right)\right), \quad \text{for } i = 0, \ldots, d-1. \tag{1}$$

This operation maps $\boldsymbol{z}$ onto a discrete coordinate $(\hat{z}_0, \ldots, \hat{z}_{d-1})$ on a regular grid. The d-dimensional coordinate can be flattened into a scale via a bijective mapping, yielding a codebook with $|\mathcal{C}| = \prod_{i=0}^{d-1} L_i$ discrete codes.

To capture fine-grained motion dynamics, R-FSQ performs recursive residual quantization over $V+1$ FSQ stages. Let $\mathbf{R}^0 = \mathbf{Z}$, the model computes a quantized approximation $\widehat{\mathbf{Z}}^v$ and updates the residual at each stage v:

$$\widehat{\mathbf{Z}}^v = \mathrm{FSQ}\left(\mathbf{R}^v\right), \quad \mathbf{R}^{v+1} = \mathbf{R}^v - \widehat{\mathbf{Z}}^v, \quad \text{for } v = 0, \ldots, V. \tag{2}$$

The final quantized representation is obtained by aggregating all hierarchical approximations: $\widehat{\mathbf{Z}} = \sum_{v=0}^{V} \widehat{\mathbf{Z}}^v$. R-FSQ is optimized end-to-end via a reconstruction objective:

$$\mathcal{L}_1 = \left\| \mathbf{X} - D\left(\widehat{\mathbf{Z}}\right) \right\|_2^2. \tag{3}$$

The R-FSQ converts the motion sequence $\mathbf{X}$ into $V+1$ ordered discrete token sequences $\{\boldsymbol{m}^0, \boldsymbol{m}^1, \ldots, \boldsymbol{m}^V\}$, where $\boldsymbol{m}^v \in \{0, \ldots, |\mathcal{C}| - 1\}^t$. This yields a coarse-to-fine discrete representation: $\boldsymbol{m}^0$ encodes global motion patterns, while higher levels refine fine-grained dynamics.

4.2 Scalable Masked Transformer

Multi-modal Condition Encoders. As shown in Fig. 6b, we use modality-specific encoders to process heterogeneous inputs, including text, audio, and motion trajectories. We adopt the pre-trained T5-XL encoder [8] to extract semantic representations from text. For audio, we use the encoder from WavTokenizer [16] to capture features from both speech and music domains. For motion trajectories, we employ a lightweight convolutional encoder to capture local temporal dependencies. All modality-specific features are projected into a common c-dimensional embedding space for unified conditioning. Formally, the encoded features are represented as $Z_{\text{text}} \in \mathbb{R}^{N_t \times c}$, $Z_{\text{audio}} \in \mathbb{R}^{N_a \times c}$ and $Z_{\text{traj}} \in \mathbb{R}^{N_{tr} \times c}$, where N_t, N_a and N_{tr} denote the sequence length of respective modality.

Parallel Mask modeling. Following the masked motion modeling paradigm in MoMask [13], we formulate motion generation as a masked token reconstruction task, where the model learns to recover original motion sequences from partially corrupted inputs. To enable scalable generation, we adopt a LLaMA-based bidirectional Transformer backbone, which supports global context reasoning beyond conventional autoregressive approaches. Built upon R-FSQ, each motion sequence $\mathbf{X}$ is represented

as $V+1$ parallel streams of discrete tokens $\{\boldsymbol{m}^0, \boldsymbol{m}^1, \ldots, \boldsymbol{m}^V\}$. This multi-stream representation introduces the challenge of jointly modeling hierarchical residual structure. To this end, we propose **parallel masked modeling strategy** that enables simultaneous encoding and prediction across all token streams, as shown in Fig. 6b.

Specifically, we implement a *consistent* masking scheme across the $V+1$ streams by randomly replacing a subset of tokens with a special $<\mathrm{MASK}>$ token. Under this scheme, once a temporal index is selected, the tokens at that timestep across all residual levels are masked simultaneously. Let $\widetilde{\boldsymbol{m}}^v$ denote the masked token sequence at the v-th residual level. To process these multi-stream tokens in parallel, we introduce a set of independent embedding layers $\{\mathrm{Embd}^v(\cdot)\}_{v=0}^V$ that map each $\widetilde{\boldsymbol{m}}^v$ into a shared latent space. The comprehensive motion representation is then obtained by aggregating embeddings across all streams:

$$Z_{\mathrm{enc}} = \sum_{v=0}^{V} \mathrm{Embd}^v\left(\widetilde{\boldsymbol{m}}^v\right), \tag{4}$$

which is subsequently fed into the Transformer backbone for contextual modeling.

For parallel token prediction, we employ $V+1$ prediction heads $\{\mathrm{FFN}^v(\cdot)\}_{v=0}^V$, each parameterized by an independent feed-forward network. Given the latent representation h produced from the transformer, each head reconstructs its respective token stream as $\widehat{\boldsymbol{m}}^v = \mathrm{FFN}^v(h)$.

Objective Function. The over Masked Transformer is optimized via a cross-entropy loss applied independently to each token stream:

$$\mathcal{L}_2 = -\sum_{v=0}^{V} \log p\left(\boldsymbol{m}^v | Z_{\mathrm{text}}, Z_{\mathrm{audio}}, Z_{\mathrm{traj}}, \widetilde{\boldsymbol{m}}^v\right) \tag{5}$$

4.3 Training Paradigm

Conventional motion generation frameworks typically rely on strictly paired datasets for training. However, in large-scale data collection, motion sequences are often not perfectly synchronized with audio signals (e.g., music or speech), as data collection prioritizes scene diversity and motion variability. As a result, high-quality audio–motion aligned data account for only about one-tenth of the OmniHuMo Dataset ($\sim$ 500 hours). To address this limitation, we adopt a staged training curriculum to enable robust cross-modal alignment under weakly aligned data.

Stage I: Text-to-motion Pre-training. We first train the model on text-to-motion task using all text-motion pairs in OmniHuMo. The audio and trajectory encoders are frozen. This stage learns a strong motion prior and aligns textual features with a structured motion representation space, providing a stable foundation for subsequent multimodal integration.

Stage II: Multi-modal Alignment. We then introduce audio-aligned data while freezing the text encoder and Transformer backbone, updating only the audio and trajectory encoders. This stage maps audio and trajectory features into the text-aligned latent space, enabling a unified cross-modal representation.

Stage III: Joint Multi-Modal Fine-tuning. Finally, we fine-tune the full model for arbitrary multimodal conditioning. To address the imbalance between text-only data and audio-motion aligned data, we use disproportional sampling: 10% of text-only data and full audio-aligned subset are used per epoch. We further apply modality augmentation by randomly injecting trajectory input for text-conditioned samples and textual input for audio-conditioned samples with probability 0.1. This improves robustness and supports flexible multimodal conditioning.

5 Experiments

We summarize the experimental results in this section. Due to space limitations, additional implementation details and evaluation metrics are provided in Appendix B, with more ablation study in

Table 2: Reconstruction performance under different data scales.

Data Size	FID ↓	MPJPE ↓
0.05M	160.65	94.55
0.6M	43.18	44.65
3M	**17.32**	**27.92**

Table 3: Ablation of different motion token modeling strategies. AR means auto-regressive prediction.

Strategy	Methods	FID ↓	R@1 ↑	MMDist ↓
A	AR-Flatten	26.71	0.62	17.17
B	Mask-Flatten	20.89	0.66	**16.54**
C	Mask-Parallel	**19.46**	**0.66**	16.78

Table 4: Motion reconstruction performance comparison measured in MPJPE (mm).

Method	Train Dataset	Test Dataset		
		HumanML3D [14]	MotionMillion [11]	OmniHuMo
ScaMo [29]	MotionUnion [29]	63.3	88.9	-
GoToZero [11]	MotionMillion [11]	41.9	45.5	36.1
Ours (R-FSQ)	OmniHuMo	**27.9**	**21.5**	**13.2**

Appendix C.3 and qualitative results in Appendix D. Comparisons with existing methods on public benchmarks are provided in Appendix C.1.

5.1 Experimental Setup

Datasets. We conduct experiments on HumanML3D [14], and our OmniHuMo dataset. OmniHuMo provides semantically aligned multimodal annotations. For the text-conditioned subset (**OmniHuMo-Text**), we split the data into 50K test, 10K validation, and the rest for training. For speech and music modalities (**OmniHuMo-Speech** and **OmniHuMo-Music**), we use 8K test, 2K validation, and the rest for training. For ablation studies, we use HumanML3D [14], a widely used 3D human motion-language benchmark. It contains 14,616 motion clips and 44,970 text descriptions (28.59 hours), with each clip annotated by 3–4 captions. We follow the standard 80%/5%/15% train/validation/test split.

Implementation Details. The motion tokenizer adopts a 4-layer Residual FSQ with a codebook size of 2048 per layer. The encoder–decoder follows SnapMoGen [12], combining convolutional residual blocks and self-attention [53], with a temporal downsampling factor of 4. It is trained with a learning rate of 2e-4 for 200 epochs on 16 NVIDIA H20 GPUs, with a batch size of 256 per GPU. The AnyMo network is built upon LLaMA [50]. To study scaling laws with respect to model capacity, we train models ranging from 111M to 3B parameters. AnyMo is trained for 210 epochs with an initial learning rate of 2×10^{-4}, decayed to 1×10^{-5} after 500 warm-up steps using cosine scheduling. Training is performed on 48 NVIDIA H20 GPUs with a batch size of 16 per GPU.

Evaluation Metrics. For motion reconstruction, we use Mean Per Joint Position Error (MPJPE) to measure geometric accuracy. For text-driven motion generation, following T2M-GPT [64], we report Motion–Text Retrieval Precision (R-Precision), Fréchet Inception Distance (FID), Multimodal Distance (MMDist), and Motion Diversity (Div). For speech-driven gesture and music-driven dance generation, following LoM [6], we use FID, Beat Alignment Score (BAS), and Div.

5.2 Ablation Study

Data scale for motion tokenizer. We study the effect of data scale on R-FSQ tokenizer. For reconstruction, the tokenizer is trained on OmniHuMo and evaluated on HumanML3D. As shown in Tab. 2, reconstruction performance consistently improves as the training data scale increases, demonstrating the importance of large-scale data for learning accurate motion representations.

Motion token modeling strategy. We compare different motion token modeling strategies in Tab. 3. Both training and evaluation are conducted on HumanML3D. First, comparing Strategy A and B, masked modeling outperforms autoregressive modeling, highlighting the benefit of bidirectional context. Second, comparing Strategy B and C, the parallel modeling strategy further improves gener-

Table 5: Comparison of text-driven motion generation performance on OmniHuMo-Text test set.

Method	FID ↓	R@1 ↑	R@2 ↑	R@3 ↑	MMDist ↓	Div →
Real	-	0.74	0.88	0.93	25.75	46.59
AnyMo-111M	262.10	0.63	0.76	0.82	29.16	43.57
AnyMo-343M	216.26	0.67	0.80	0.86	28.23	44.98
AnyMo-775M	148.81	0.71	0.83	0.88	27.24	45.36
AnyMo-1B	102.21	0.74	0.86	0.90	26.28	45.74
AnyMo-3B	**55.59**	**0.75**	**0.87**	**0.91**	**25.87**	**46.71**

Table 6: Comparison of speech-driven motion generation performance on OmniHuMo-Speech test set.

Method	FID ↓	BAS ↑	Div →
Real	-	0.205	44.40
AnyMo-111M	178.83	0.204	42.54
AnyMo-343M	201.01	0.202	42.00
AnyMo-775M	**83.80**	0.205	42.86
AnyMo-1B	96.87	0.208	42.66
AnyMo-3B	91.12	**0.214**	**43.47**

Table 7: Comparison of music-driven motion generation performance on OmniHuMo-Music test set.

Method	FID ↓	BAS ↑	Div →
Real	-	0.210	39.22
AnyMo-111M	70.98	0.207	39.78
AnyMo-343M	74.18	0.209	**39.13**
AnyMo-775M	**34.41**	0.210	38.22
AnyMo-1B	37.62	0.211	36.99
AnyMo-3B	46.17	**0.213**	38.15

ation quality. Moreover, compared to the flattening strategy, the parallel design enables simultaneous decoding of multiple token streams, improving computational efficiency.

5.3 Motion Reconstruction Comparison

We compare motion reconstruction performance of tokenizers trained on different datasets in Tab. 4, we evaluate three methods: our R-FSQ tokenizer trained on OmniHuMo, ScaMo's FSQ [29] trained on MotionUnion, and GoToZero's tokenizer [11] trained on MotionMillion. All models are evaluated on the test sets of HumanML3D [14], MotionMillion [11], and OmniHuMo. As shown in Tab. 4, our method consistently achieves the best reconstruction performance across all benchmarks. This comparison evaluates the combined effects of tokenizer design and training data. To examine tokenizer design under the same training data, we provide a controlled comparison in Appendix C.2.

5.4 Single-Modality Motion Generation Performance

Text-driven Motion Generation. Following prior work [33], we train a motion-text retrieval model on the OmniHuMo-Text training set and evaluate AnyMo on its test split. As shown in Tab. 5, performance improves consistently as the model scales from 111M to 3B parameters, indicating stable gains under the current data and training regime, consistent with empirical scaling laws.

Audio-driven Motion Generation. We evaluate *Speech-to-Gesture* and *Music-to-Dance* on OmniHuMo-Speech and OmniHuMo-Music test sets, respectively. As shown in Tab. 6 and Tab. 7, audio-driven generation does not exhibit monotonic scaling behavior. While AnyMo-775M achieves the best FID, further increasing model capacity leads to slight performance degradation, suggesting potential overfitting under limited paired audio–motion data. In contrast, Beat Alignment Score (BAS) improves with model size, indicating stronger temporal modeling and better cross-modal synchronization between audio rhythm and motion.

Table 8: Comparison of multi-modal conditional inputs in text-driven motion generation task. Evaluated on OmniHuMo-Text test set using AnyMo-3B.

Text	Traj	FID ↓	R@1 ↑	Trajectory Error (>50cm) (%) ↓	Location Error (>50cm) (%) ↓	Average Error (cm) ↓
✔		55.59	0.75	0.52	0.28	50.50
✔	✔	**41.43**	**0.77**	**0.33**	**0.14**	**27.16**

Table 9: Comparison of multi-modal conditional inputs in speech-driven motion generation task.

Speech	Text	Traj	FID ↓	BAS ↑	Div →
✔(Real)			-	0.205	44.40
✔			91.12	0.214	43.47
✔	✔		89.74	0.215	**43.63**
✔		✔	76.82	0.217	42.51
✔	✔	✔	**76.55**	**0.217**	43.18

Table 10: Comparison of multi-modal conditional inputs in music-driven motion generation task.

Music	Text	Traj	FID ↓	BAS ↑	Div →
✔(Real)			-	0.210	39.22
✔			46.17	0.213	**38.15**
✔	✔		43.84	0.213	37.34
✔		✔	**42.99**	0.214	37.93
✔	✔	✔	43.26	**0.215**	36.94

5.5 Multi-Modality Motion Generation Performance

Text-driven Motion Generation. Tab. 8 reports results under multi-modal conditioning for text-driven motion generation. Incorporating auxiliary modalities consistently improves performance across metrics. In particular, adding trajectory information enhances motion realism (FID), retrieval accuracy (R@1), and reduces trajectory error. These results indicate that trajectory cues provide useful structural priors for improving both motion quality and text alignment.

Audio-driven Motion Generation. Tab. 9 and Tab. 10 present results for speech- and music-driven motion generation under multi-modal conditioning. Overall, incorporating additional modalities improves performance. Compared to text, trajectory conditioning yields larger gains in motion realism (FID) and beat alignment (BAS), highlighting stronger structural guidance. However, motion diversity does not consistently increase with more modalities, likely due to the reduced flexibility of the motion space under additional conditioning constraints.

6 Conclusion

In this paper, we introduce **OmniHuMo**, the first large-scale human motion dataset with rich multi-modal annotations, providing a foundation for multimodal motion modeling. Based on OmniHuMo, we propose **AnyMo**, a unified framework for controllable motion generation from arbitrary combinations of text, speech, music, and trajectories. Experiments show that our method achieves high-quality and diverse motion synthesis with flexible multimodal control. These results highlight the importance of large-scale, well-aligned data for improving generalization and controllability. We hope OmniHuMo and AnyMo will advance general-purpose human motion generation and multimodal generative modeling.

Acknowledgments and Disclosure of Funding

This work is partially supported by the National Natural Science Foundation of China (NSFC) under Grant Nos. 62461160331, 62376259, 62306301, and 62576334.

References

[1] Shuai Bai, Yuxuan Cai, Ruizhe Chen, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

[2] Zhongang Cai, Wanqi Yin, Ailing Zeng, Chen Wei, Qingping Sun, Yanjun Wang, Hui En Pang, Haiyi Mei, Mingyuan Zhang, Lei Zhang, Chen Change Loy, Lei Yang, and Ziwei Liu. SMPLer-X: Scaling up expressive human pose and shape estimation. In *Advances in Neural Information Processing Systems*, volume 36, pages 11454–11468, 2023.

[3] Bin Cao, Sipeng Zheng, Hao Luo, Boyuan Li, Jing Liu, and Zongqing Lu. Opent2m: No-frill motion generation with open-source, large-scale, high-quality data. *arXiv preprint arXiv:2603.18623*, 2026.

[4] Zhi Cen, Huaijin Pi, Sida Peng, Zehong Shen, Minghui Yang, Shuai Zhu, Hujun Bao, and Xiaowei Zhou. Generating human motion in 3d scenes from text descriptions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1855–1866, 2024.

[5] Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T Freeman. Maskgit: Masked generative image transformer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11315–11325, 2022.

[6] Changan Chen, Juze Zhang, Shrinidhi K Lakshmikanth, Yusu Fang, Ruizhi Shao, Gordon Wetzstein, Li Fei-Fei, and Ehsan Adeli. The language of motion: Unifying verbal and non-verbal language of 3d human motion. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 6200–6211, 2025.

[7] Yafeng Chen, Siqi Zheng, Hui Wang, Luyao Cheng, Tinglong Zhu, Rongjie Huang, Chong Deng, Qian Chen, Shiliang Zhang, Wen Wang, et al. 3d-speaker-toolkit: An open-source toolkit for multimodal speaker verification and diarization. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2025.

[8] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53, 2024.

[9] Joon Son Chung and Andrew Zisserman. Out of time: Automated lip sync in the wild. In *Computer Vision – ACCV 2016 Workshops*, pages 251–263. Springer, 2017.

[10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[11] Ke Fan, Shunlin Lu, Minyue Dai, Runyi Yu, Lixing Xiao, Zhiyang Dou, Junting Dong, Lizhuang Ma, and Jingbo Wang. Go to zero: Towards zero-shot motion generation with million-scale data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13336–13348, 2025.

[12] Chuan Guo, Inwoo Hwang, Jian Wang, and Bing Zhou. Snapmogen: Human motion generation from expressive texts. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

[13] Chuan Guo, Yuxuan Mu, Muhammad Gohar Javed, Sen Wang, and Li Cheng. Momask: Generative masked modeling of 3d human motions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1900–1910, 2024.

[14] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5152–5161, 2022.

[15] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3.6M: Large scale datasets and predictive methods for 3D human sensing in natural environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(7):1325–1339, 2014.

[16] Shengpeng Ji, Ziyue Jiang, Wen Wang, Yifu Chen, Minghui Fang, Jialong Zuo, Qian Yang, Xize Cheng, Zehan Wang, Ruiqi Li, et al. Wavtokenizer: an efficient acoustic discrete codec tokenizer for audio language modeling. *arXiv preprint arXiv:2408.16532*, 2024.

[17] Tao Jiang, Xinchen Xie, and Yining Li. Rtmw: Real-time multi-person 2d and 3d whole-body pose estimation. *arXiv preprint arXiv:2407.08634*, 2024.

[18] Rahima Khanam and Muhammad Hussain. Yolov11: An overview of the key architectural enhancements. *arXiv preprint arXiv:2410.17725*, 2024.

[19] Hui Li, Mingwang Xu, Yun Zhan, Shan Mu, Jiaye Li, Kaihui Cheng, Yuxuan Chen, Tan Chen, Mao Ye, Jingdong Wang, et al. Openhumanvid: A large-scale high-quality dataset for enhancing human-centric video generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 7752–7762, 2025.

[20] Jiefeng Li, Jinkun Cao, Haotian Zhang, Davis Rempe, Jan Kautz, Umar Iqbal, and Ye Yuan. Genmo: A generalist model for human motion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11766–11776, 2025.

[21] Ronghui Li, YuXiang Zhang, Yachao Zhang, Hongwen Zhang, Jie Guo, Yan Zhang, Yebin Liu, and Xiu Li. Lodge: A coarse to fine diffusion network for long dance generation guided by the characteristic dance primitives. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1524–1534, 2024.

[22] 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.

[23] 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.

[24] Zhuo Li, Mingshuang Luo, Ruibing Hou, Xin Zhao, Hao Liu, Hong Chang, Zimo Liu, and Chen Li. Morph: A motion-free physics optimization framework for human motion generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14580–14589, 2025.

[25] Zeyu Ling, Bo Han, Shiyang Li, Jikang Cheng, Hongdeng Shen, and Changqing Zou. Versatilemotion: A unified framework for motion synthesis and comprehension. *arXiv preprint arXiv:2411.17335*, 2024.

[26] Haiyang Liu, Naoya Iwamoto, Zihao Zhu, Zhengqing Li, You Zhou, Elif Bozkurt, and Bo Zheng. Disco: Disentangled implicit content and rhythm learning for diverse co-speech gestures synthesis. In *Proceedings of the 30th ACM international conference on multimedia*, pages 3764–3773, 2022.

[27] Haiyang Liu, Zihao Zhu, Giorgio Becherini, Yichen Peng, Mingyang Su, You Zhou, Xuefei Zhe, Naoya Iwamoto, Bo Zheng, and Michael J Black. Emage: Towards unified holistic co-speech gesture generation via expressive masked audio gesture modeling. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1144–1154, 2024.

[28] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: A skinned multi-person linear model. *ACM Transactions on Graphics*, 34(6):248:1–248:16, 2015.

[29] Shunlin Lu, Jingbo Wang, Zeyu Lu, Ling-Hao Chen, Wenxun Dai, Junting Dong, Zhiyang Dou, Bo Dai, and Ruimao Zhang. Scamo: Exploring the scaling law in autoregressive motion generation model. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 27872–27882, 2025.

[30] Naureen Mahmood, Nima Ghorbani, Nikolaus F Troje, Gerard Pons-Moll, and Michael J Black. AMASS: Archive of motion capture as surface shapes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5442–5451, 2019.

[31] Fabian Mentzer, David Minnen, Eirikur Agustsson, and Michael Tschannen. Finite scalar quantization: Vq-vae made simple. *arXiv preprint arXiv:2309.15505*, 2023.

[32] Wenshuo Peng, Kaipeng Zhang, and Sai Qian Zhang. T3M: Text guided 3D human motion synthesis from speech. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 1168–1177. Association for Computational Linguistics, June 2024.

[33] Mathis Petrovich, Michael J Black, and Gül Varol. Temos: Generating diverse human motions from textual descriptions. In *European Conference on Computer Vision*, pages 480–497. Springer, 2022.

[34] Ekkasit Pinyoanuntapong, Muhammad Usama Saleem, Korrawe Karunratanakul, Pu Wang, Hongfei Xue, Chen Chen, Chuan Guo, Junli Cao, Jian Ren, and Sergey Tulyakov. MaskControl: Spatio-temporal control for masked motion synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9955–9965, 2025.

[35] Ekkasit Pinyoanuntapong, Muhammad Usama Saleem, Pu Wang, Minwoo Lee, Srijan Das, and Chen Chen. Bamm: Bidirectional autoregressive motion model. *arXiv preprint arXiv:2403.19435*, 2024.

[36] Ekkasit Pinyoanuntapong, Pu Wang, Minwoo Lee, and Chen Chen. Mmm: Generative masked motion model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1546–1555, 2024.

[37] Matthias Plappert, Christian Mandery, and Tamim Asfour. The KIT motion-language dataset. *Big Data*, 4(4):236–252, dec 2016.

[38] Xingqun Qi, Yatian Wang, Hengyuan Zhang, Jiahao Pan, Wei Xue, Shanghang Zhang, Wenhan Luo, Qifeng Liu, and Yike Guo. Co^{3}Gesture: Towards coherent concurrent co-speech 3D gesture generation with interactive diffusion. In *The Thirteenth International Conference on Learning Representations*, 2025.

[39] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021.

[40] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision. In *International conference on machine learning*, pages 28492–28518. PMLR, 2023.

[41] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. Sam 2: Segment anything in images and videos. arXiv preprint arXiv:2408.00714, 2024.

[42] Davis Rempe, Mathis Petrovich, Ye Yuan, Haotian Zhang, Xue Bin Peng, Yifeng Jiang, Tingwu Wang, Umar Iqbal, David Minor, Michael de Ruyter, et al. Kimodo: Scaling controllable human motion generation. *arXiv preprint arXiv:2603.15546*, 2026.

[43] Isaac Robinson, Peter Robicheaux, Matvei Popov, Deva Ramanan, and Neehar Peri. Rf-detr: Neural architecture search for real-time detection transformers. arXiv preprint arXiv:2511.09554, 2025.

[44] Simon Rouard, Francisco Massa, and Alexandre Défossez. Hybrid transformers for music source separation. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2023.

[45] Zehong Shen, Huaijin Pi, Yan Xia, Zhi Cen, Sida Peng, Zechen Hu, Hujun Bao, Ruizhen Hu, and Xiaowei Zhou. World-grounded human motion recovery via gravity-view coordinates. In *SIGGRAPH Asia 2024 Conference Papers*, pages 1–11, 2024.

[46] 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.

[47] Tomás Soucek and Jakub Lokoc. Transnet v2: An effective deep network architecture for fast shot transition detection. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 11218–11221, 2024.

[48] Zachary Teed and Jia Deng. DROID-SLAM: Deep Visual SLAM for Monocular, Stereo, and RGB-D Cameras. *Advances in neural information processing systems*, 2021.

[49] Guy Tevet, Sigal Raab, Brian Gordon, Yonatan Shafir, Daniel Cohen-Or, and Amit H. Bermano. Human motion diffusion model. arXiv preprint arXiv:2209.14916, 2022.

[50] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[51] 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.

[52] Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

[53] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

[54] Weilin Wan, Zhiyang Dou, Taku Komura, Wenping Wang, Dinesh Jayaraman, and Lingjie Liu. Tlcontrol: Trajectory and language control for human motion synthesis. In *European Conference on Computer Vision*, pages 37–54. Springer, 2024.

[55] Ye Wang, Sipeng Zheng, Bin Cao, Qianshan Wei, Weishuai Zeng, Qin Jin, and Zongqing Lu. Scaling large motion models with million-level human motions. *arXiv preprint arXiv:2410.03311*, 2024.

[56] Yuxin Wen, Qing Shuai, Di Kang, Jing Li, Cheng Wen, Yue Qian, Ningxin Jiao, Changhai Chen, Weijie Chen, Yiran Wang, et al. Hy-motion 1.0: Scaling flow matching models for text-to-motion generation. *arXiv preprint arXiv:2512.23464*, 2025.

[57] Haoning Wu, Erli Zhang, Liang Liao, Chaofeng Chen, Jingwen Hou, Annan Wang, Wenxiu Sun, Qiong Yan, and Weisi Lin. Exploring video quality assessment on user generated contents from aesthetic and technical perspectives. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 20144–20154, 2023.

[58] Lixing Xiao, Shunlin Lu, Huaijin Pi, Ke Fan, Liang Pan, Yueer Zhou, Ziyong Feng, Xiaowei Zhou, Sida Peng, and Jingbo Wang. Motionstreamer: Streaming motion generation via diffusion-based autoregressive model in causal latent space. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10086–10096, October 2025.

[59] Yiming Xie, Varun Jampani, Lei Zhong, Deqing Sun, and Huaizu Jiang. Omnicontrol: Control any joint at any time for human motion generation. *arXiv preprint arXiv:2310.08580*, 2023.

[60] Haofei Xu, Jing Zhang, Jianfei Cai, Hamid Rezatofighi, Fisher Yu, Dacheng Tao, and Andreas Geiger. Unifying flow, stereo and depth estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(11):13941–13958, 2023.

[61] Ziyue Yang, Kaixing Yang, and Xulong Tang. TokenDance: Token-to-token music-to-dance generation with bidirectional mamba. *arXiv preprint arXiv:2603.27314*, 2026.

[62] Hongwei Yi, Hualin Liang, Yifei Liu, Qiong Cao, Yandong Wen, Timo Bolkart, Dacheng Tao, and Michael J Black. Generating holistic 3d human motion from speech. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 469–480, 2023.

[63] Zebin You, Shen Nie, Xiaolu Zhang, Jun Hu, Jun Zhou, Zhiwu Lu, Ji-Rong Wen, and Chongxuan Li. Llada-v: Large language diffusion models with visual instruction tuning. *arXiv preprint arXiv:2505.16933*, 2025.

[64] Jianrong Zhang, Yangsong Zhang, Xiaodong Cun, Shaoli Huang, Yong Zhang, Hongwei Zhao, Hongtao Lu, and Xi Shen. T2m-gpt: Generating human motion from textual descriptions with discrete representations. *arXiv preprint arXiv:2301.06052*, 2023.

[65] Jinlu Zhang, Zixi Kang, and Yizhou Wang. Opendance: Multimodal controllable 3d dance generation using large-scale internet data. *arXiv preprint arXiv:2506.07565*, 2025.

[66] Yuang Zhang, Tiancai Wang, and Xiangyu Zhang. Motrv2: Bootstrapping end-to-end multi-object tracking by pretrained object detectors. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 22056–22065, 2023.

[67] Yuhong Zhang, Jing Lin, Ailing Zeng, Guanlin Wu, Shunlin Lu, Yurong Fu, Yuanhao Cai, Ruimao Zhang, Haoqian Wang, and Lei Zhang. Motion-x++: A large-scale multimodal 3d whole-body human motion dataset. *arXiv preprint arXiv:2501.05098*, 2025.

[68] Zeyu Zhang, Yiran Wang, Wei Mao, Danning Li, Rui Zhao, Biao Wu, Zirui Song, Bohan Zhuang, Ian Reid, and Richard Hartley. Motion anything: Any to motion generation. *arXiv preprint arXiv:2503.06955*, 2025.

[69] Fengqi Zhu, Rongzhen Wang, Shen Nie, Xiaolu Zhang, Chunwei Wu, Jun Hu, Jun Zhou, Jianfei Chen, Yankai Lin, Ji-Rong Wen, et al. Llada 1.5: Variance-reduced preference optimization for large language diffusion models. *arXiv preprint arXiv:2505.19223*, 2025.

Appendix

Appendices A–D provide supplementary dataset descriptions, experimental configurations, quantitative results, and visualizations, respectively. Appendix E discusses limitations and future work.

A Dataset Information

A.1 Related Work

Data-driven Motion Modeling. Motion synthesis quality heavily depends on the scale and diversity of training data. Traditional optical motion capture datasets, such as AMASS [30] and HumanML3D [14], are constrained by high acquisition costs and limited variety. Recent efforts have explored automatic motion extraction from in-the-wild videos. For example, MotionX++ [67] proposes an automated pipeline that leverages pose estimation techniques [2] to extract motion sequences from large-scale internet videos. Building on this paradigm, [11, 55, 3, 56] introduce million-scale motion datasets, marking significant progress toward large-scale motion modeling. However, these datasets primarily focus on text-conditioned generation and lack comprehensive multimodal annotations. To address this gap, we propose an efficient pipeline for harvesting motion with precisely aligned multimodal annotations from web videos, enabling scalable multimodal motion generation.

A.2 Details of OmniHuMo Construction

This section describes the filtering procedures for raw videos and 3D motion annotations, as illustrated in Fig. S1. We apply strict filtering criteria, resulting in 3.2M motion sequences distilled from over 200M raw videos. We also present the prompts used for data captioning.

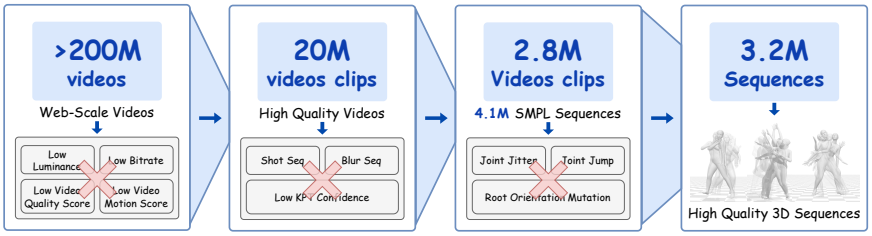


Figure S1: Filter operators in the data processing pipeline.

A.2.1 Video Filtering.

The collected source videos contain a substantial amount of low-quality content. To ensure reliable downstream annotation, we apply a series of filtering criteria.

Average bitrate. Bitrate reflects information density and serves as a proxy for visual quality. We compute the normalized average bitrate as $B/\sqrt{W \times H}$, where W, H and B denote resolution and bitrate. Videos with a normalized bitrate below 500 are discarded.

Luminance. Following [19], we measure overall video brightness to filter extreme lighting conditions. The luminance score is computed as $0.2126R + 0.7152G + 0.0722B$, where R, G, and B denote RGB values. Videos with luminance outside $[10, 210]$ are removed.

Video quality score. We first apply frame-level CLIP [39] aesthetic scoring for efficient pre-filtering, removing videos with an average score below 4.0. We then adopt DOVER [57] for comprehensive video quality assessment combining aesthetic and technical cues. Videos with a final score below 0.25 are filtered out.

Motion score. High-motion scenes often introduce severe motion blur that degrades annotation accuracy, while overly static scenes provide limited informative motion cues. To address this, we estimate optical flow using UniMatch [60] and compute the average flow magnitude as a motion score. Videos with scores outside $[3.5, 350]$ are discarded.

A.2.2 Human 3D Annotation

We employ GVHMR [45] to reconstruct 3D human motion in world coordinates. It estimates poses in a Gravity-View (GV) coordinate system defined by gravity and camera viewing directions, which reduces ambiguity in world coordinate definition. The model takes bounding boxes, 2D keypoints, video frames, and relative camera rotations as inputs, and predicts SMPL parameters, including root translation t, body pose θ, root rotation r, and shape parameters β.

To obtain camera motion, we employ DROID-SLAM [48] to estimate camera extrinsics. Since dynamic human regions degrade SLAM accuracy, we follow prior work to mask moving humans during Dense Bundle Adjustment. In practice, SAM2 [41] masks are unstable under fast motion. We instead use RF-DETR [43] detection boxes to mask dynamic regions, which improves the stability and accuracy of camera estimation.

To reduce instability in 3D reconstruction caused by occlusions and camera estimation errors, we further filter reconstructed SMPL sequences using the following criteria.

Root Orientation Mutation. Although GVHMR is robust, abrupt root orientation changes may still occur under rapid camera motion or occlusion. We measure the rotation difference between consecutive frames:

$$\Delta\theta_i = \arccos\left(\frac{\mathrm{tr}(R_i R_{i-1}^\top) - 1}{2}\right), \tag{6}$$

where R_i denotes the root rotation at frame i. Sequences with $\Delta\theta_i > 30^\circ$ are discarded.

Joint Jitter. We quantify temporal smoothness using jerk. For joint j at frame i with position $\mathbf{p}_{i,j}$, we compute:

$$\dddot{\mathbf{J}}_{i,j} = \mathbf{p}_{i+1,j} - 3\mathbf{p}_{i,j} + 3\mathbf{p}_{i-1,j} - \mathbf{p}_{i-2,j}, \tag{7}$$

The average jerk over the sequence is:

$$\dddot{\mathbf{J}} = \frac{1}{(T-3)J} \sum_{i=3}^{T-1} \sum_{j=1}^{J} \| \dddot{J}_{i,j} \|_2. \tag{8}$$

Sequences with $\dddot{\mathbf{J}} > 0.015$ are discarded.

Joint Position Jump. To detect sudden local motion anomalies, we compute the maximum frame-wise joint displacement:

$$\mathrm{Jump}_i = \max_j \|\mathbf{p}_{i,j} - \mathbf{p}_{i-1,j}\|_2 . \tag{9}$$

Sequences with Jump_i above 200mm are removed.

A.2.3 Motion Caption Annotation

We use Qwen3-VL-32B [1] to generate fine-grained action descriptions within detected human bounding boxes. The prompt design is shown in Fig. S2. To ensure quality, we constrain the model to describe only the target person's actions and poses, excluding irrelevant information such as clothing, facial attributes, background, camera motion, or other unobservable details. Each motion sequence is annotated with 1–3 captions, each limited to 30 words. Captions are required to use precise action verbs (e.g., "standing on right leg," "swinging left arm") and temporal connectors (e.g., "then," "while") to form coherent action descriptions. For specific categories such as dance and sports, we additionally encourage explicit activity labels (e.g., "lat pulldown," "Latin dance") alongside the description.

You are a video-based human motion annotation system specialized in analyzing human movement patterns. Analyze the motion of the person enclosed by the bounding box in the given video and generate motion caption.

ANALYSIS SCOPE:
- Analyze ONLY the person action and motion enclosed by the bounding box
- DO NOT describe clothing (color, type, texture), expression, bounding boxes and camera movement
- DO NOT describe background objects unless they are directly being interacted with the bounded person
- DO NOT fabricate any details that are not directly observable.
- DO NOT repeat motion content across captions.

OUTPUT REQUIREMENTS:
1. Generate 1-3 motion caption, containing no more than 30 words.
2. Use temporal connectors (then, while, following) to link sub-motions into one continuous narrative. For example, "The person walks forward, then bends the upper body forward and reaches down with the right hand to pick up an object".
3. Use precise verbs (e.g., "Sitting cross-legged", "Wave left hand" .etc) rather than generic ones (e.g., "moves").
4. If the person is performing a noticeable action, such as rowing, playing golf, stand on one's head, etc., please add a description of the corresponding action, for example: The person is performing a golf swing, with a backswing followed by a powerful forward swing.
5. If the person is interacting with an object, please describe the interaction in detail, for example: The person is sitting on a chair, then stands up and walks forward to pick up a box, holding it with both hands.
6. If the person is dancing, output a general description of the dance style, for example: This person is dancing ballet, lifting both hands over their head, turning their body, and the tips of their feet are lifted up.
7. If no movement is detected, return an empty list: {"motion_caption": [] }.

When writing motion captions, you can focus on one or more of the following aspects for each description:
1. Body movements: Involved joints or limbs, posture changes, movement speed, and motion trajectories. For example, "The person raises their right arm above their head, then quickly lowers it while stepping forward with the left foot".
2. Global movement: Body orientation, movement direction, gait characteristics, and high-level action types (e.g., walking, running, jumping). For example, "The person is walking forward with a slight lean to the left, taking long strides and swinging their arms".

Return ONLY valid JSON with this exact structure: {"motion_caption": [string1, string2, ...]}

Figure S2: Prompt design for text description annotation in OmniHuMo.

B Experimental Configuration

B.1 Training Setup

Data Pipeline. The data construction pipeline is deployed across multiple clusters. Video curation runs on a CPU cluster, while the Human 2D & 3D and audio annotation are processed on 100 L20 GPUs. Motion captioning is performed on a separate 40 H20 GPUs cluster. Overall, the pipeline generates approximately 100k high-quality motion sequences per day.

Motion Tokenizer. The motion tokenizer adopts a residual FSQ architecture with 4 layers and a codebook size of 2048 per layer. The encoder and decoder follows SnapMoGen [12], consisting of alternating convolutional residual blocks and self-attention [53], with temporal downsampling by a factor of 4. It is trained on OmniHuMo using AdamW with an initial learning rate of 2×10^{-4}. A multi-step decay is applied at epochs $[60, 140]$ with a factor of 0.3, for 200 epochs. Training uses 16 NVIDIA H20 GPUs with a batch size of 256 per GPU.

AnyMo Training. The AnyMo network is built upon the LLaMA architecture [50], with RMSNorm applied before attention and feed-forward layers. To study scaling behavior, we train models ranging from 111M to 3B parameters. Optimization uses AdamW with an initial learning rate of 2×10^{-4}, 500 warm-up steps, and cosine decay to 1×10^{-5}. Training runs for 210 epochs on 48 NVIDIA H20 GPUs with a batch size of 16 per GPU.

B.2 Evaluation Metrics.

Motion Reconstruction. We use Mean Per Joint Position Error (MPJPE) to measure geometric accuracy, computed as the average L2 distance between reconstructed and ground-truth joint positions across all frames.

Text-driven Motion Generation. Following T2M-GPT [64], we evaluate text-driven motion generating using FID, R-Precision, Div, and MMDist:

- **FID**: Fréchet Inception Distance measures the distribution gap between generated and real motions, computed as Fréchet distance between their feature distributions in embedding space.
- **R-Precision**: Motion–Text Retrieval Precision measures the alignment between generated motions and input text via retrieval accuracy. For each motion, we rank its Euclidean distances to 32 candidate text descriptions (1 ground truth and 31 randomly ssampled negatives) and report Top 1/2/3 retrieval accuracy.
- **Div**: Diversity is computed as the average pairwise Euclidean distance between randomly sampled motion features, reflecting the spread of generated samples.
- **MMDist**: MultiModel Distance measures the average Euclidean distance between motion features and their corresponding text feature, indicating cross-modal alignment quality.

Speech-driven Gesture Generation. Following LoM [6], we use FID, BAS and Div to evaluate gesture generation performance. FID and Div are computed in the same way as in text-driven motion generation. Beat Alignment Score (BAS) measures temporal synchronization between audio beats and generated motion beats. It is computed as the average Gaussian-weighted alignment between each audio beat and its nearest motion beat based on squared temporal distance.

Music-driven Dance Generation. We use the same evaluation metrics as in speech-driven gesture generation, including FID, BAS, and Div. Traditional dance generation [46] uses five metrics, including FID_k and Div_k for kinematic feature distribution and diversity, FID_g and Div_g for geometric feature distribution, and BAS for motion–music synchronization. However, EDGE [51] shows that these kinematic and geometric metrics are unreliable due to heuristic feature design that fails to capture high-level semantics. Therefore, we adopt a contrastive learning-based feature extractor and report FID, BAS, and Div.

C Experimental Results

C.1 Generation on Public Benchmarks

We train AnyMo on the public dataset used by the compared methods: HumanML3D [14] for text-to-motion, FineDance [22] for music-to-dance, and BEATv2 [27] for speech-to-gesture. This setting examines generation performance separately from the scale of OmniHuMo. On HumanML3D, AnyMo achieves the lowest FID and the highest R@1 and R@3 among the compared methods (Tab. S1).

On FineDance, AnyMo achieves the highest kinematic diversity and improves over EDGE in FID_k and BAS, while Lodge and TokenDance achieve lower FID_k and higher BAS (Tab. S2). Here, FID_k and Div_k use kinematic features and differ from the learned-feature FID and Div in the OmniHuMo experiments.

Table S1: Text-to-motion generation on HumanML3D.

Method	FID ↓	R@1 ↑	R@3 ↑
GT	–	0.702	0.914
MoMask [13]	12.232	0.621	0.846
MotionStreamer [58]	11.790	0.631	0.859
AnyMo (Ours)	**7.8096**	**0.7158**	**0.919**

Table S2: Music-to-dance generation on FineDance using kinematic feature metrics and BAS.

Method	FID_k ↓	Div_k ↑	BAS ↑
EDGE [51]	94.34	8.13	0.2116
Lodge [21]	**45.56**	6.75	**0.2397**
TokenDance [61]	47.20	6.62	0.2385
AnyMo (Ours)	55.4920	**8.5240**	0.2301

For BEATv2, we retain the body component of EMAGE predictions and evaluate both methods with the same body-only representation and evaluator. AnyMo achieves lower FID and higher BAS in this setting (Tab. S3).

C.2 Tokenizer Comparison with Matched Training Data

Table 4 compares tokenizers trained on different datasets. To isolate tokenizer design, we retrain GoToZero [11] and R-FSQ on the same HumanML3D training split with the same motion representation and evaluation protocol. R-FSQ reduces MPJPE from 55.689 to 37.439 mm, an improvement of 18.250 mm (Tab. S4). Separately, Tab. 2 keeps the tokenizer architecture fixed to evaluate the benefit of more training data.

Table S3: Speech-to-gesture generation on BEATv2 with body-only evaluation.

Method	FID ↓	BAS ↑
EMAGE [27]	51.4396	0.2159
AnyMo (Ours)	**37.4659**	**0.2205**

Table S4: Tokenizer reconstruction on HumanML3D with matched training data and evaluation.

Tokenizer	MPJPE (mm) ↓
GoToZero [11]	55.689
R-FSQ (Ours)	**37.439**

C.3 Ablation Study

This section presents additional ablation studies on the motion tokenizer R-FSQ. For the reconstruction task, the motion tokenizer is trained on a 200K subset of OmniHuMo and evaluated on HumanML3D [14]. For the motion generation task, both training and evaluation are conducted on HumanML3D.

Codebook size on R-FSQ. We study the effect of codebook size on the reconstruction performance, as shown in Tab. S5. The results indicate that reconstruction and generation quality improve consistently with larger codebooks. Notably, increasing training data yields more significant gains than scaling the codebook size alone.

Number of residual layers on R-FSQ. Tab. S6 provides an ablation study on the number of residual layers, evaluated on both reconstruction and generation performance. As shown in Tab. S6, reconstruction quality improves with increasing residual depth. However, generation performance degrades when the number of residual layers exceeds 4, likely due to increased prediction difficulty introduced by additional token streams that must be modeled jointly.

Table S5: Comparison of reconstruction and generation performance under different data scales.

Codebook Size	Reconstruction		Generation (T2M)	
	FID ↓	MPJPE ↓	FID ↓	R@1 ↑
1024	101.86	76.05	34.67	0.59
2048	77.35	69.69	30.80	0.60
4096	72.24	65.71	30.26	0.60
8192	73.30	65.38	31.36	0.59
16384	**71.63**	**64.93**	**28.52**	**0.60**

Table S6: Comparison of reconstruction and generation performance under different residual depths.

Residual Depth	Reconstruction		Generation (T2M)	
	FID ↓	MPJPE ↓	FID ↓	R@1 ↑
1	101.86	76.05	33.01	0.61
2	81.64	61.54	25.32	0.62
4	52.80	49.06	**19.46**	**0.66**
6	45.90	41.47	19.93	0.65
8	**38.58**	**34.79**	23.64	0.60

D Visualizations

D.1 OmniHuMo Examples

We present visualization examples of OmniHuMo in Fig. S3, S4, and S5. These results demonstrate that OmniHuMo covers diverse motion patterns with strong multimodal alignment, providing a high-quality foundation for large-scale motion modeling.

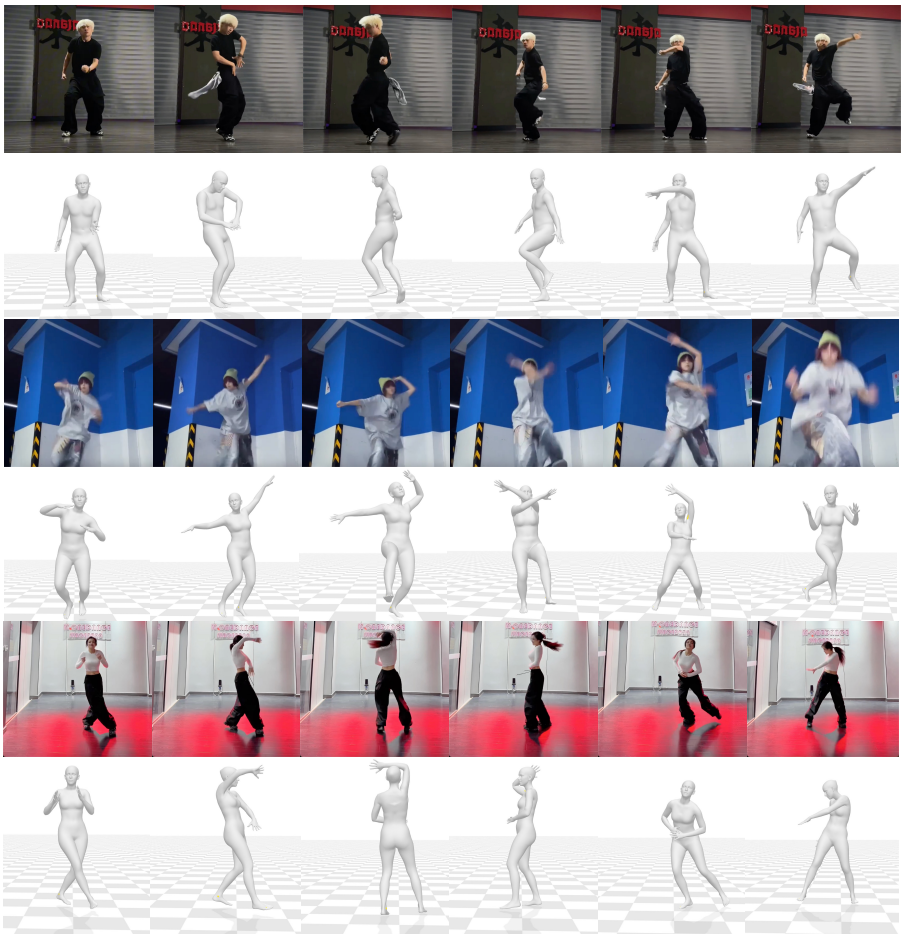


Figure S3: Visualization of 3D SMPL reconstructions for motion sequences in OmniHuMo.

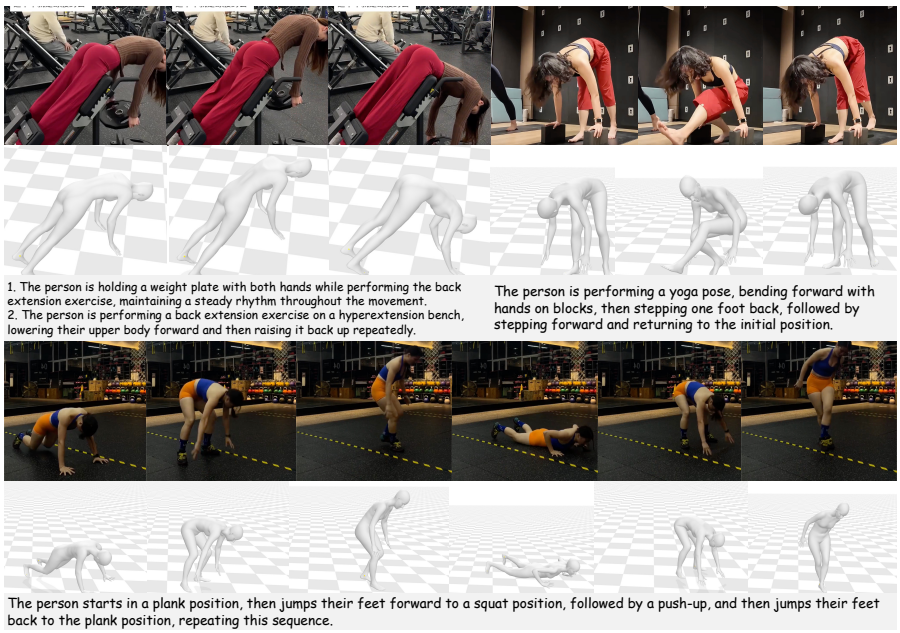


Figure S4: Visualization of SMPL reconstruction and the corresponding text description.

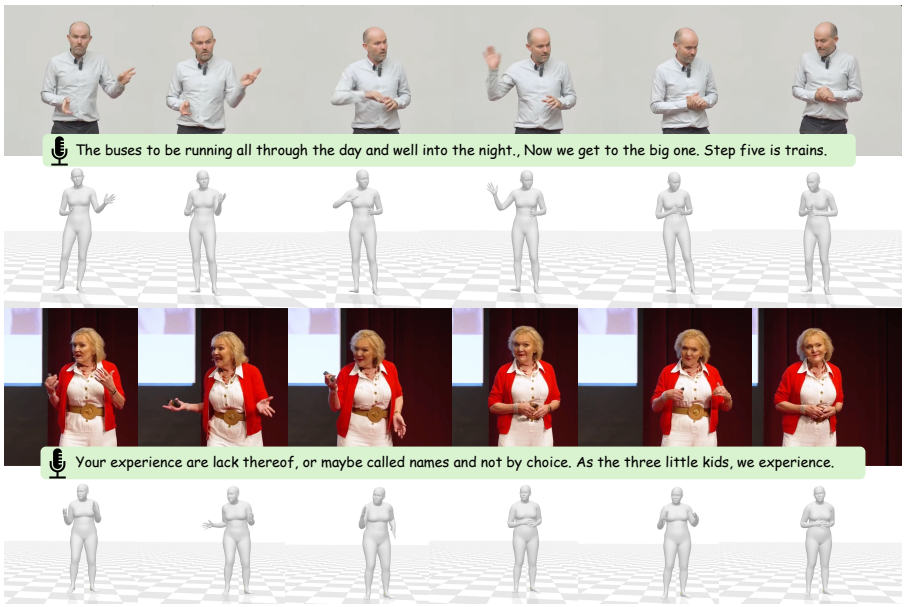


Figure S5: Visualization of the speaker's SMPL reconstruction and the corresponding transcript.

D.2 AnyMo Generation Examples

We present additional examples to visualize motions generated by AnyMo, as shown in Fig. S6, S7, and S8. For trajectory-controlled generation, following TLControl [54], we adopt a test-time optimization strategy to refine coarse predictions for more precise trajectory control. These visualization results demonstrate that our model produces motion sequences that closely follow diverse input modalities.

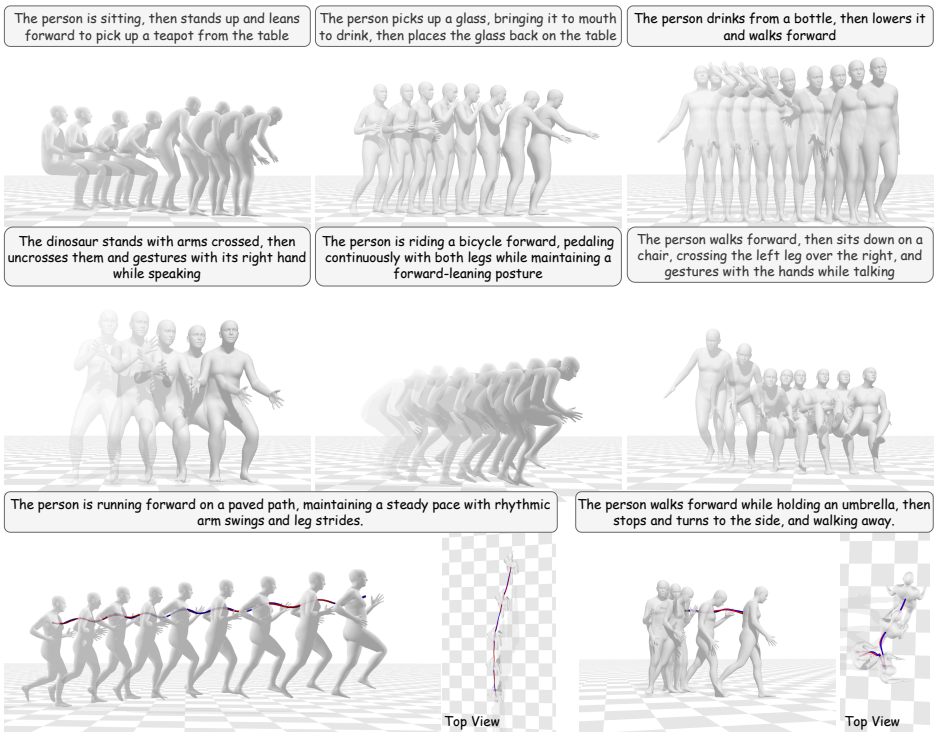


Figure S6: Visualization on text-driven motion generation task.

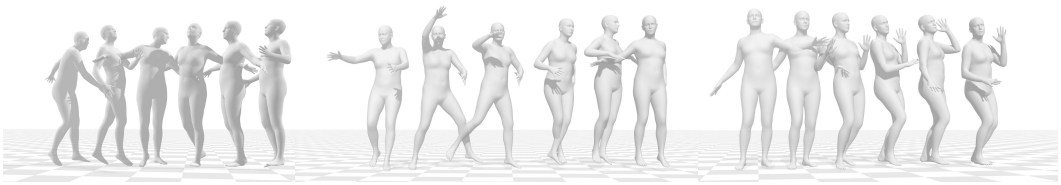


Figure S7: Visualization on music-driven motion generation task.

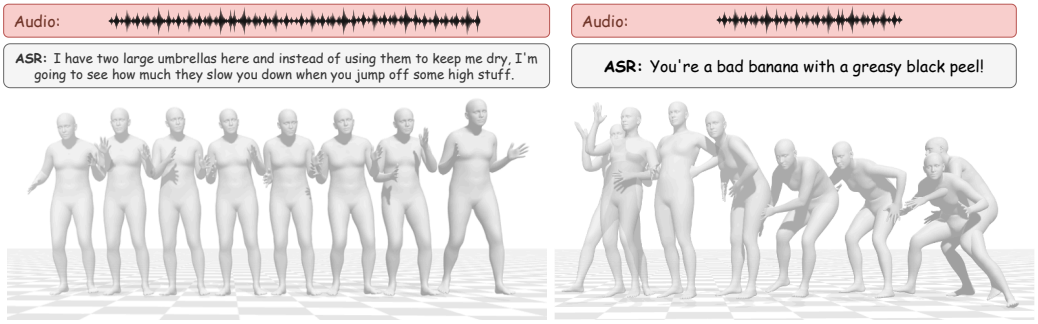


Figure S8: Visualization on speech-driven motion generation task.

E Limitations and Future Work

Despite the advances demonstrated in this work, several limitations remain and open avenues for future research. First, OmniHuMo does not include finger joint annotations. Compared with body movements, hand regions in internet videos are more frequently affected by occlusion and motion blur, resulting in extremely low data usability for reliable hand reconstruction. Second, audio-aligned data accounts for only a limited portion of the dataset. Future work could improve data diversity and coverage through more targeted data collection strategies.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: A summary of the paper's contributions is provided in the introduction.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Limitations are discussed in the Appendix E.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

Justification: Not Applicable.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide detailed efforts to ensure reproducibility. See Section 5 and Appendix C.3 for dataset descriptions, evaluation metrics and implementation details.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We will release the code upon acceptance to facilitate reproduction.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: See Section 5 and Appendix C.3 for dataset and implementation details.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [N/A]

   Justification: Not Applicable.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: See Section 5 and Appendix C.3 for implementation details.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discuss broader impacts in Section 6.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The proposed models and datasets do not pose a high risk for misuse. Furthermore, strict filtering criteria were applied during dataset construction.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: See References

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: Not Applicable.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: Not Applicable.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: Not Applicable.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: The paper describes the use of Qwen3-VL-32B for generating fine-grained action descriptions in Section 3.1 (Motion Caption Annotation) and provides prompt details in Appendix A.2.3.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.