

KinMo: Kinematic-aware Human Motion Understanding and Generation

Supplementary Document

A. Overview

This supplementary document contains further details on dataset collection and our framework, additional ablation studies and experimental results, and limitations of KinMo. For more visual results, refer to the demo videos on the project page: <https://andypinxinliu.github.io/KinMo>.

B. Dataset Collection Details

Unlike conventional dataset annotation approaches, which require extensive human labeling and verification, our semi-automatic data collection pipeline with human in the loop significantly reduces the cost while providing high-quality annotation. The procedure is described in the following:

Pose Text Description Generation. To generate text descriptions for individual poses, we adopt PoseScript [2], which provides detailed descriptions for each pose by capturing fine-grained details of joint positions and their spatial relationships. In addition to providing a textual representation of the pose, PoseScript is capable of describing the relative positions of different joints, which is crucial for understanding complex human motion.

Keyframe Selection. While PoseScript offers per-frame annotations, it does not provide a direct way to capture the temporal transitions between poses over time. We observe that text descriptions for temporally adjacent frames are often very similar. In contrast, frames that are farther apart in time exhibit less overlap in their descriptions, often presenting different semantics.

Based on this observation, we devise a keyframe-based approach detailed as follows. We utilize sBERT [16] to extract embeddings for the PoseScript-generated descriptions of each frame. By calculating the cosine similarity between these text embeddings, we can measure the similarity of poses across frames. If the cosine similarity between two frames falls below a threshold of 0.8, we classify the frame as a keyframe, marking a significant temporal transition. This allows us to isolate key moments in the sequence that represent meaningful pose changes and filter out redundant frames for subsequent analysis. We then compute temporal local motions by analyzing kinematic group differences across a specified time window, allowing us to capture finer

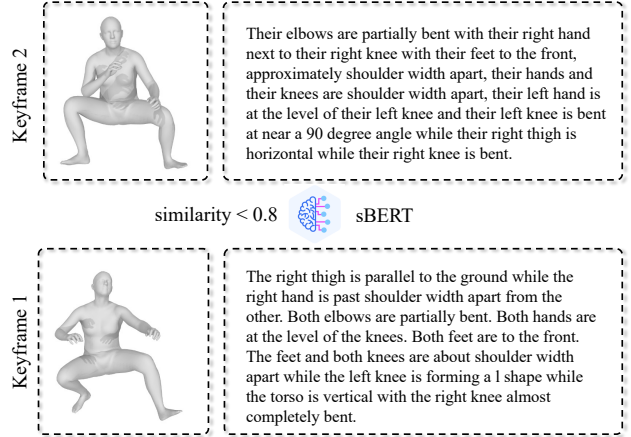


Figure 1. **Visualization of Sample Keyframes.** We use PoseScript to obtain per-frame text annotations for poses and select keyframes based on sentence similarities.

motion details within the pose sequence. Fig. 1 shows a visualization of sample keyframes within our dataset.

Temporal Joint Motion Reasoning. For reasoning about the potential motions of kinematic groups across keyframes, we leverage a Large Language Model (LLM), specifically GPT-4 [11]. Given the text descriptions of the selected keyframes, GPT-4 is tasked with generating potential motions for each kinematic group by reasoning through the temporal relationships between poses. To guide the LLM, we provide a well-structured prompt design, shown in Tab. 1. We give a system prompt to use GPT-4 as a motion labeler, providing a specific answer format instruction and content instruction. During each query, the user prompt contains one specific annotation example with explanation for one-shot in-context-learning and the keyframe descriptions needed to be annotated. A real example prompt for a single motion sequence is provided in Tab. 8, and its corresponding annotation result is shown in Tab. 9. This design allows the model to infer the most likely motions of specific kinematic groups and predict how they may evolve over time. The reasoning process enables the system to generate coherent and realistic motion sequences based on the selected keyframes.

Human Evaluation. Human evaluators play a critical role in ensuring the accuracy and quality of text and motion pre-

Table 1. **Prompt template for automatic dataset annotation.** We show one specific example in Tab. 8 with more details.

<p>System Prompt</p> <p>You are a motion description labeler who should describe the motion using your language as detailed as possible. Now, describe the motion in the given video or text by describing the motion of each kinematic group respectively: Kinematic groups: torso, neck, left arm, right arm, left leg, right leg.</p> <p>[Answer Format Instruction]</p> <p>To describe a motion, you should describe Inside each group and Between groups: INDIVIDUAL GROUP: torso: <MOTION DESCRIPTION>; neck: <MOTION DESCRIPTION>...</p> <p>[Answer Content Instruction]</p> <p>In your description, you can use simple adjectives or numerical scale (distance/degree/speed) to describe each motion (for example, push forward for 3 meters, from 0 to 45 degrees, etc)...</p> <p>User Prompt</p> <p>[One-shot Example for In-Context-Learning]</p> <p>Think about the motion: [a man stumbles to his right. the motion seems sudden so he was probably pushed. a person standing loses balance, falling to the right and recovers standing...</p> <p>[Data to be Annotated]</p>
--

dictionaries generated by the system. The evaluation process is conducted in two stages:

- **Keyframe Selection Evaluation.** Two human evaluators review the selected keyframes by the system and the corresponding rendered motion sequences. They identify the optimal cosine similarity threshold for filtering keyframes and examine whether the selected keyframes adequately capture the most significant pose transitions. The threshold is iteratively adjusted based on the evaluators’ feedback to improve the precision of keyframe selection.
- **Text Description Evaluation.** The evaluators also assess the generated text descriptions for each joint-level motion. They determine whether the descriptions accurately reflect the motion dynamics and satisfy the intended criteria. If errors or inconsistencies are found, the evaluators provide feedback to the system, prompting a revision of the current prompt design. This iterative process continues until the evaluators reach a consensus, measured by a Cohen’s Kappa statistic of at least 0.8, indicating a strong agreement.

The evaluators spent about one day interacting with LLM for the iterative prompt optimization. Both the average

Table 2. **Evaluation Results of KinMo detailed descriptions.** Bad Response Rate (BRR) is assessed by $BRR = \frac{\text{items with score} < 5}{\text{total items}}$.

Evaluation Method	preservation of spatial detail	capture of temporal dynamics	consistency with the global text	Metrics
Human evaluation	8.24 2.96%	8.01 3.15%	8.71 1.32%	Average Score BRR
LLM evaluation	8.30 2.68%	8.12 2.37%	8.55 1.55%	Average Score BRR

prompt length and returning answer is less than 1000 words, resulting in approximately 3200 tokens for both input and output, and **23 USD** final cost for the whole annotation (44,970 motion sequences).

Accuracy of KinMo Annotation. We conducted two complementary evaluations: (a) *Human evaluation*: Five independent annotators (not involved in the annotation) assessed 500 randomly selected samples. Each annotator viewed the motion video and scored the associated description on a scale of 0–10 for three criteria: spatial accuracy, temporal coherence, and consistency with the global text. The average scores in Tab. 2 indicate strong alignment between motion and text. (b) *LLM-based evaluation*: We used GPT-4o-mini [11] to perform the same evaluation, using a structured prompt (Tab. 1) and scoring based on the same criteria. LLM-based scores were similarly high, with $\leq 5\%$ of samples flagged as potentially inconsistent. These evaluations demonstrate that KinMo’s auto-generated descriptions reliably capture both spatial and temporal motion details at scale, reinforcing the quality of our dataset beyond indirect task-based metrics.

C. Additional Implementation Details

Hierarchical Text-Motion Alignment. Both the motion and text encoders are based on Transformer architectures [19]. We add two tokens in front of the raw sequence to represent the mean and standard deviation, akin to those in the VAE-based ACTOR model [12]. These encoders are probabilistic, generating parameters of a Gaussian distribution (μ and Σ) from which a latent vector $z \in \mathbb{R}^d$ can be sampled. The text encoder processes input features from a pretrained and frozen RoBERTa [9] model, while the motion sequence is given directly as input to the motion encoder. As for the cross-attention that connects three levels of semantics, we use one transformer block containing one multi-head attention layer with one MLP layer. For the neural network, we set the latent dimension to 512, the number of heads to 6, and the feed-forward size to 1024. We train this module for 70 epochs, with other settings the same as in TMR [13].

Text-Motion Generation. We use MoMask [6] as our generator architecture. During the training process, to enhance the model’s robustness to variations in text input, we randomly omit 10% of the text conditioning. This approach

Table 3. **Global Action Text, Low-level Text and Motion as Tri-modality Retrieval Benchmark on HumanML3D [4]**. HText denotes Global Action-level descriptions, LText demotes joint-level motion descriptions.

Setting	HText-Motion Retrieval						Motion-HText Retrieval					
	R@1 ↑	R@2 ↑	R@3 ↑	R@5 ↑	R@10 ↑	MedR ↓	R@1 ↑	R@2 ↑	R@3 ↑	R@5 ↑	R@10 ↑	MedR ↓
(a) All	3.67	7.17	10.32	15.73	25.12	40.00	4.39	8.08	11.56	17.23	26.81	38.00
(b) All with threshold	7.98	13.87	18.47	25.86	36.39	22.00	7.60	12.27	16.40	21.97	31.36	30.00
(c) Dissimilar subset	34.15	52.44	58.54	72.56	81.10	2.00	37.20	54.88	62.80	68.90	79.27	2.00
(d) Small batches	60.76	75.79	81.35	86.93	91.79	1.10	61.26	76.26	81.97	87.24	91.63	1.11
Setting	LText-Motion Retrieval						Motion-LText Retrieval					
	R@1 ↑	R@2 ↑	R@3 ↑	R@5 ↑	R@10 ↑	MedR ↓	R@1 ↑	R@2 ↑	R@3 ↑	R@5 ↑	R@10 ↑	MedR ↓
(a) All	3.57	7.17	9.82	14.77	24.30	37.00	4.15	8.17	11.39	15.89	25.42	37.00
(b) All with threshold	7.12	12.39	16.65	23.85	35.48	22.00	7.31	12.06	16.06	21.09	31.21	30.00
(c) Dissimilar subset	45.36	70.10	75.26	80.41	85.57	2.00	46.39	68.04	75.26	81.44	86.60	2.00
(d) Small batches	62.21	78.29	83.97	88.57	93.08	1.08	63.10	77.98	83.87	88.74	93.15	1.05
Setting	HText-LText Retrieval						LText-HText Retrieval					
	R@1 ↑	R@2 ↑	R@3 ↑	R@5 ↑	R@10 ↑	MedR ↓	R@1 ↑	R@2 ↑	R@3 ↑	R@5 ↑	R@10 ↑	MedR ↓
(a) All	0.05	0.10	0.12	0.17	0.38	1905.0	1.72	3.00	4.34	6.41	10.89	194.0
(b) All with threshold	0.07	0.12	0.17	0.57	1.12	1544.0	3.19	5.50	7.48	10.60	16.42	132.0
(c) Dissimilar subset	1.03	2.06	4.12	6.19	15.46	48.00	22.68	36.08	45.36	55.67	72.16	4.00
(d) Small batches	4.34	7.82	11.78	19.44	37.05	14.77	40.51	55.56	64.69	76.10	89.07	2.14

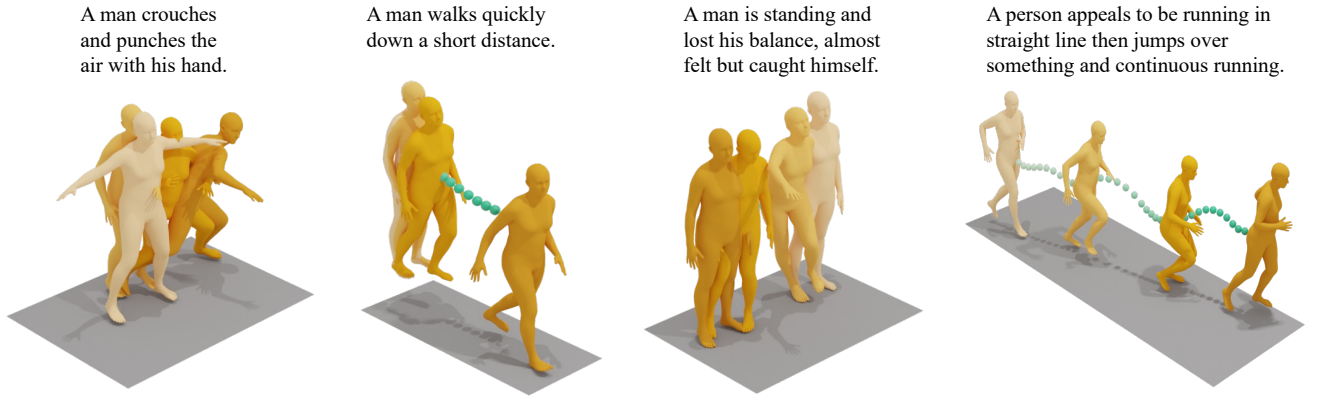


Figure 2. **Visualization of Motion Trajectory Control.** We leverage pelvis locations to guide the motion generation in addition to the text.

also facilitates the use of Classifier-Free Guidance (CFG). Our codebook consists of 512 entries, with each having a 512-dimensional embedding and 6 residual layers. The Transformer’s embedding size is 384 and has 6 attention heads, each with an embedding dimension of 64, spread across 8 layers. Both the encoder and decoder reduce the motion sequence length by a factor of 4 when transitioning to the token space. The learning rate follows a linear warm-up schedule, peaking at $2e-4$ after 2000 iterations. We utilize AdamW optimizer. The mini-batch size is 512 during

the training of RVQ-VAE and 64 for training the Transformers. At inference, the CFG scale is set to $cfg = 4$ for the base layer and $cfg = 5$ for the 6 residual layers, with the generation process running for 10 iterations. To produce text embeddings, we apply Hierarchical Text-Motion Alignment (HTMA), which results in embeddings of size 512. These embeddings are subsequently reprojected to a 384-dimensional space to match the Transformer’s token size.

Motion Editing. We leverage a *Joint Motion Reasoner* to refine both global and local action descriptions using the

Motion Trajectory Control

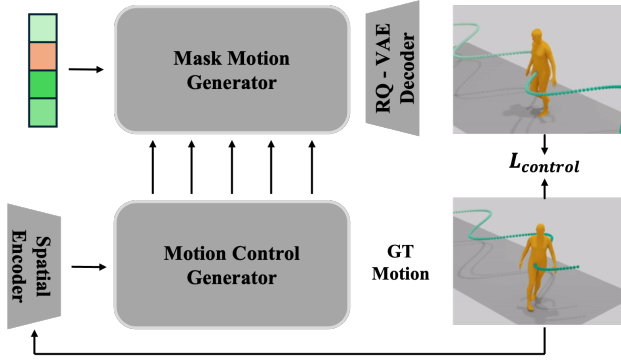


Figure 3. **Motion Trajectory Control.** We adopt a ControlNet architecture to condition the generator with the provided trajectory of the target joint during the generation. We utilize a CNN encoder to process the spatial position information and feed it as the input condition into the control generator network.

users’ input. This model enables precise action-level edits (e.g., changing *running* to *jumping*) or local joint adjustments (e.g., *slightly raising the hands*). Our method follows a coarse-to-fine approach, assisted by a masking mechanism, to perform these edits at varying levels of granularity. Specifically, by masking the target sequences and using the mask generator to fill in the masked area, we can dynamically adjust the motion to meet the target requirement. For more details on the masking-based editing process, please refer to MMM [15].

Motion Trajectory Control. Inspired by ControlNet [21] for Diffusion Models, we incorporate the joint spatial conditioning for trajectory control. As shown in Fig.3, during this stage, the motion control model is a trainable replica of the frozen mask motion generator. Specifically, each layer in the motion Control Generator is appended with a zero-initialized linear layer to remove random noise in the initial training steps. The initial τ poses are defined by the trajectories of K control joints, $\mathbf{g}^{1:\tau} = \{\mathbf{g}^i\}_{i=1}^{\tau}$, where $\mathbf{g}^i \in \mathbb{R}^{K \times 3}$ denotes the global absolute locations of each control joint. A Trajectory Encoder Θ^b consisting of convolution layers is used to encode the trajectory signals. Unlike previous methods like OmniControl [20, 21], which directly diffuses in the motion space to allow for explicit supervision of control signals, effectively supervising control signals in the latent space is non-trivial. Therefore, in addition to using a motion reconstruction loss based on RQ-Tokenzer decoder \mathcal{D} to decode the latent $\hat{\mathbf{z}}_0$ into the motion space, we also add a control loss $\mathcal{L}_{\text{control}}$ to obtain the predicted motion $\hat{\mathbf{x}}_0$:

$$\mathcal{L}_{\text{control}} = \mathbb{E} \left[\frac{\sum_i \sum_j m_{ij} \|R(\hat{\mathbf{x}}_0)_{ij} - R(\mathbf{x}_0)_{ij}\|_2^2}{\sum_i \sum_j m_{ij}} \right], \quad (1)$$

where $R(\cdot)$ converts the joint local positions to global absolute locations and $m_{ij} \in \{0, 1\}$ is the binary joint mask at frame i for the joint j .

In our network design, *Motion Control Generator* is a trainable copy of *Masked Transformer* with the zero linear layer connected to the output of each Masked Transformer layer in MoMask [6] to mitigate random noise in the initial training steps. The spatial encoder is a 3-layer CNN-based Residual network with a temporal downsampling of factor 4 to encode the trajectory control signal (the joint position information to be controlled along the sequence).

D. Additional Experimental Results

D.1. Text-Motion Alignment

In this work, beyond the hierarchical text-semantics framework proposed in the main paper, we explored group- and interaction-level text as alternative semantic representations for motion. Specifically, we investigated tri-modal alignment between global action text, low-level text (including both group-level and interaction-level descriptions), and motion. This initial approach was intuitive. As our joint-motion reasoner is capable of generating group-level motion and group interaction scripts conditioned on action-level motion descriptions, we explored the understanding capabilities of different semantic levels of motion during modality alignment.

Challenges with Text Modality Alignment. As shown in Tab. 3, we observe that retrieval performance between global action text and low-level text is significantly worse compared to retrieval between text and motion modalities. We attribute this to the limitations of existing text encoders, which fail to capture the necessary reasoning capabilities to align the corresponding motions at the joint- or global-action-level. Unlike the joint-motion reasoner in the main paper, which leverages large language models (LLMs) to model these relationships, the text encoders used here do not have the same capacity for motion inference. Interestingly, we find that low-level text to global action text retrieval performs better than global action to low-level retrieval. We hypothesize that this occurs because low-level descriptions are more specific, directly corresponding to joint-level motion patterns, whereas global action descriptions are often more ambiguous. For instance, *running* can correspond to a wide variety of motion sequences (e.g., running with hands raised or running with hands at the sides), making it more challenging to align with specific low-level motion details.

Text-Motion Retrieval at Different Levels. As shown in Tab. 3, low-level text descriptions exhibit better alignment with the motion modality compared to global action descriptions. This is because low-level text is more directly tied to specific motion patterns, describing precise joint

Table 4. **Cross-Attention Order of Descriptions for Text-Motion Retrieval.**

Semantic Sequence	Text-to-motion retrieval				Motion-to-text retrieval			
	R@1 ↑	R@2 ↑	R@3 ↑	MedR ↓	R@1 ↑	R@2 ↑	R@3 ↑	MedR ↓
group-inter-global	7.98	12.18	15.64	24.00	8.28	12.47	19.24	21.00
global-inter-group	8.97	14.01	18.92	18.00	8.93	13.99	20.12	18.00
global-group-inter	9.05	15.23	20.47	16.00	9.01	15.92	21.42	16.00

movements, while global action descriptions are more abstract and can encompass multiple motion sequences. The increased ambiguity of global action text makes it harder to align with motion data, which further explains the observed discrepancy in retrieval performance.

Description Integration Order for Text-Motion Alignment. As shown in Tab. 4, we switched the integration order of global action, group-level descriptions, and interaction-level descriptions for the text-motion alignment process. We observe that even though the order largely affects the performance¹, all the proposed strategies with additional descriptions outperform previous methods, confirming that they are beneficial for the alignment, with an extra performance boost using a coarse-to-fine approach.

D.2. Coarse-to-Fine Motion Generation

To assess the contribution of each component within our pipeline, we design the following variations: (1) CLIP-C: Only global motion description (original HumanML3D text) is applied for motion generation, akin to MoMask [6]; (2) CLIP-G: We add group-level semantics using CLIP for motion generation; and (3) CLIP-I: We additionally add interaction-level semantics to CLIP-G for motion generation. We also apply these three settings for Hierarchical Text Motion Alignment (HTMA) to validate the effectiveness of our coarse-to-fine generation strategy and the benefits of text-motion alignment for motion generation. Fig. 4 shows that a coarse-to-fine procedure can enhance the motion generation quality. In addition, our proposed text-motion alignment can significantly speed up training and boost performance.

User Study Details. We recruited 20 participants with good English proficiency to evaluate randomly selected 80 videos from each method: MoMask [6], MMM [15], STMC [14], and ours. Participants were never informed of the source of the videos for a fair assessment. Fig. 5 shows a snapshot of the user study website.

D.3. Text-Motion Editing

Evaluation Metrics. Due to the lack of benchmark datasets and metrics, we generate 200 fine-grained text-prompts with their corresponding edited version using GPT4-o [11]. The comparison is conducted by utilizing models to first

¹The global-group-interaction approach performs best on the retrieval task.

generate the motion corresponding to the original text and then do editing to this generation based on the new instruction. To evaluate the editing quality, beyond generation metrics, we propose using Text-Motion Similarity score to measure the similarity of edited motion with editing global motion description, denoted as HTMA-S.

Evaluation Results. We benchmark KinMo against various methods for T2M generation [6, 14, 15]. KinMo is the only method able to do local temporal editing, while maintaining organic motion generation, as shown in the main paper and Tab. 5. Editing global semantics can be captured at both the joint and interaction semantic levels, thus achieving better generation and editing. Please refer to the experiments in the main paper.

D.4. Motion Trajectory Control

Evaluation Metrics. Beyond the metrics shown in the main paper, we also include three additional metrics: (1) *Trajectory error* (Traj. err.): measures the ratio of unsuccessful trajectories, characterized by any control joint location error exceeding a predetermined threshold; (2) *Location error* (Loc. err.): represents the ratio of unsuccessful joints; and (3) *Average error* (Avg. err.): denotes the mean location error of the control joints.

Evaluation Results. We compare KinMo with open-source models [1, 21], specifically focusing on pelvis control. For fairness in the comparison, we exclude test-time optimization for all baselines. Tab. 6 shows that our method achieves more robust and accurate controlled generation with lower errors and FID score than other methods.

Additional Evaluation Results. We extend the previous comparison by conducting experiments on all joints. Tab. 7 presents a quantitative evaluation of our method on the trajectory control of all joints, while Fig. 2 shows qualitative results.

E. Details of Metrics

Motion Quality. Frechet Inception Distance (FID) quantifies the difference between the distribution of generated and real motions, using a feature extractor specific to a given dataset, such as HumanML3D [4].

Motion Diversity. Following [3, 5, 7], we present metrics such as Diversity and MultiModality to assess the variation in generated motions. Diversity evaluates the spread of the generated motions across the entire dataset. Specifically, two subsets of equal size S_d are drawn randomly from all generated motions, along with their respective feature vectors $\mathbf{v}_1, \dots, \mathbf{v}_{S_d}$ and $\mathbf{v}'_1, \dots, \mathbf{v}'_{S_d}$. The Diversity score is given by:

$$\text{Diversity} = \frac{1}{S_d} \sum_{i=1}^{S_d} \|\mathbf{v}_i - \mathbf{v}'_i\|_2. \quad (2)$$

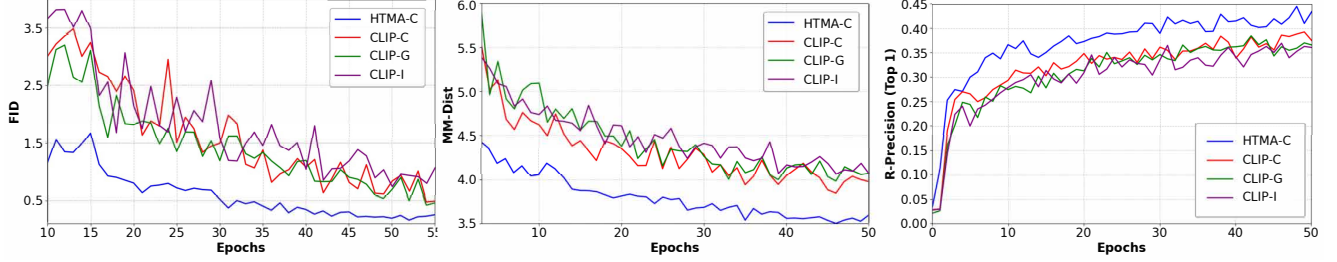


Figure 4. **Ablation for Motion Generation Process.** Our coarse-to-fine procedure helps to improve the motion generation quality. Hierarchical Text-Motion Alignment can significantly speed up the training process with better generation results and text-motion alignment.

Table 5. **Comparison of Motion Editing.** G represents control for global action, J represents control for group-level, and I represents control on interaction-level.

Methods	FID ↓	R-Prec(Top 3) ↑	MM-Dist ↓	Diversity →	HTMA-S ↑
STMC	0.561	0.612	3.864	8.952	0.636
MMM [15]	0.102	0.685	3.574	9.573	0.598
MoMask [6]	0.068	0.696	3.825	9.424	0.575
Ours (C)	0.089	0.712	3.434	9.453	0.712
Ours (C + G)	0.083	0.754	3.356	9.575	0.721
Ours (C + G + I)	0.086	0.734	3.203	9.364	0.744

Table 7. **Quantitative Results for all Joints of Trajectory Control.**

Joint	R-Prec ↑ (Top-3)	FID ↓	Traj. Err. ↓ (50 cm)	Loc. Err. ↓ (50 cm)	Avg. Err. ↓
pelvis	0.712	0.077	0.0875	0.0187	0.0787
torso	0.723	0.091	0.0933	0.0127	0.0776
left arm	0.722	0.093	0.0843	0.0132	0.0823
right arm	0.709	0.121	0.0887	0.0144	0.0814
left leg	0.707	0.084	0.0876	0.0142	0.0925
right leg	0.720	0.076	0.0828	0.0133	0.0932

MultiModality (MModality) gauges the extent of variation in motions generated from the same textual description. A set of C textual descriptions is selected at random, and then two equal-sized subsets I are chosen from the motions conditioned on the c -th description. Their feature vectors $\mathbf{v}_{c,1}, \dots, \mathbf{v}_{c,I}$ and $\mathbf{v}'_{c,1}, \dots, \mathbf{v}'_{c,I}$ are used to compute MModality as follows:

$$\text{MModality} = \frac{1}{C \times I} \sum_c c = 1^C \sum_{i=1}^I \|\mathbf{v}_{c,i} - \mathbf{v}'_{c,i}\|_2. \quad (3)$$

Condition Matching. Motion and text feature extractors provided by [4] allow for the generation of closely aligned features for matched text-motion pairs and vice versa. Within this feature space, motion-retrieval precision (R-Precision) is calculated by mixing generated motions with 31 mismatched motions, followed by computing the Top-1/2/3 text-motion matching accuracy. Multimodal Distance (MM-Dist) computes the average distance between generated motions and corresponding texts.

Control Error. As described in [21], we report the Trajectory, Location, and Average error to evaluate the preci-

Table 6. **Comparison of Motion Trajectory Control.** Here, we consider the pelvis only, excluding test-time optimization.

Methods	FID ↓	R-Precision ↑ Top 3	Traj. err. ↓ (50cm)	Loc. err. ↓ (50cm)	Avg. err. ↓
Real	0.002	0.797	0.0000	0.0000	0.0000
MDM [18]	0.698	0.602	0.4022	0.3076	0.5959
PriorMDM [17]	0.475	0.583	0.3457	0.2132	0.4417
OmniControl [21]	0.212	0.678	0.3041	0.1873	0.3226
MotionLCM [1]	0.531	0.752	0.1887	0.0769	0.1897
KinMo (Ours)	0.103	0.756	0.2034	0.0696	0.1657

sion of motion control. Trajectory error (Traj. err.) represents the fraction of failed trajectories, where a trajectory is deemed unsuccessful if a control joint in the generated motion exceeds a predefined distance threshold from the corresponding joint in the control trajectory. Similarly, Location error (Loc. err.) reflects the proportion of joints whose positions fail to meet the specified threshold. For our experiments, we utilized a 50cm threshold to compute the Trajectory and Location errors. The Average error (Avg. err.) refers to the mean distance between the control joint positions in the generated motion and their corresponding positions in the control trajectory.

F. Limitations

While our method demonstrates significant improvements over existing baselines, it still has certain limitations. The most critical limitation lies in the quality of the motion description. Low-quality motion descriptions may harm the generation performance and even worsen that of the baseline approach when no enhancement is applied to the original descriptions. Fig. 6 shows an example failure case. The reason for the generation failure is that our pipeline may miss capturing descriptions with a very short temporal span.

Ethical Considerations. While our work focuses on generating human motion videos, it raises ethical concerns due to its potential misuse for photorealistic human motion re-targeting. We emphasize the importance of responsible use and recommend implementing practices such as watermarking and deepfake detection to mitigate the risks involving deepfake videos and animated representations.

Subjective Evaluation of Human Motion Videos

Thank you for participating in the subjective evaluation.

Instructions:

Please watch each video and rate the videos based on Three evaluation metrics,
 1. Realness: How human-like the motion in the video looks
 2. Alignment: How close the motion represents to its text description
 3. Overall: Overall quality of the video
 Please rate each video on a scale of 1 to 5, where 1 is the lowest and 5 is the highest

Group 1


Reference Video	Realness Quality	Alignment Quality	Overall Quality
<p>person is walking forwards quite fast, then squats down to pick something up to then turn around and walk fast again, appears to be in a rush and moving an item</p> 	<p>1. Terrible, Completely Unnatural movements 2. Poor, with many errors and unnatural 3. Fair, hard to judge 4. Good, better, it looks real 5. Excellent, it is what a natural human motion</p> <p>○ 1 ○ 2 ○ 3 ○ 4 ○ 5</p>	<p>1. Terrible, it is not what the text describes at all 2. Poor, poorly aligned with the text description 3. Fair, it is hard to judge 4. Good, almost aligns with text, with small error 5. Excellent, it is exactly what the text describes</p> <p>○ 1 ○ 2 ○ 3 ○ 4 ○ 5</p>	<p>1. Terrible, it is not good at all 2. Poor, it is not good 3. Fair, it is hard to judge 4. Good, it is good 5. Excellent, it is perfect</p> <p>○ 1 ○ 2 ○ 3 ○ 4 ○ 5</p>

Figure 5. Screen Shot of Our User Study Website. Each user will rates the videos without knowing the source method.

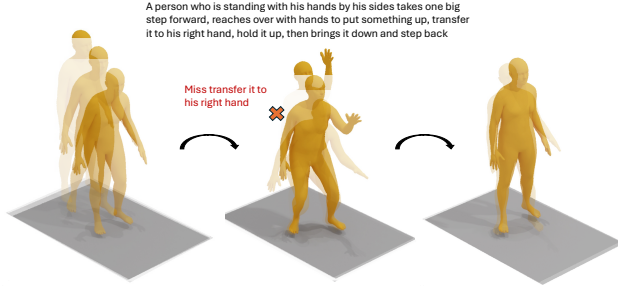


Figure 6. **Limitations.** If the text description contains many short transitions, our method may sometime miss one step.

G. Conversion of Motion Representation

G.1. Keypoint-Level Formulation

Following HumanML3D[4], a motion can be represented as the absolute and relative movement of each keypoint.

Let $J = \{1, 2, \dots, N\}$ denote the set of all keypoints in the human body model (e.g., $N = 22$ for the SMPL model). Let $T \subset \mathbb{R}$ be the continuous time domain over which the motion is defined.

For each joint $j \in J$ at time $t \in T$, we define:

1. Root Data (for the root joint, denoted as j_0):
 - Rotation Velocity: $\omega_{\text{root}}(t) \in \mathbb{R}^3$
 - Linear Velocity: $\mathbf{v}_{\text{root}}(t) \in \mathbb{R}^3$
 - Height: $h_{\text{root}}(t) \in \mathbb{R}$
2. Joint Position: $\mathbf{p}_j(t) \in \mathbb{R}^3$
3. Joint Rotation: $\mathbf{R}_j(t) \in \text{SO}(3)$, represented in a continuous 6D representation $\mathbf{r}_j(t) \in \mathbb{R}^6$
4. Joint Velocity: $\mathbf{v}_j(t) = \frac{d\mathbf{p}_j(t)}{dt} \in \mathbb{R}^3$
5. Foot Contact Information (for foot joints $j \in J_{\text{foot}} \subseteq J$): $c_j(t) \in \{0, 1\}$, where 1 indicates contact with the ground

The Keypoint-Level Formulation is then defined as the col-

lection of all these functions over time:

$$\mathcal{M} = \{(\mathbf{p}_j(t), \mathbf{R}_j(t), \mathbf{v}_j(t)) \mid j \in J, t \in T\} \cup \{\omega_{\text{root}}(t), \mathbf{v}_{\text{root}}(t), h_{\text{root}}(t) \mid t \in T\} \cup \{c_j(t) \mid j \in J_{\text{foot}}, t \in T\}.$$

G.2. Joint-Group Formulation

While the original formulation is kinematically reasonable, it often results in a many-to-many matching problem [8], making fine-grained motion descriptions based on kinematic joints difficult to express in natural language. Our proposed formulation addresses this issue by leveraging natural language to describe individual body part movements with ease, organically. For example:

- *The person is walking forward, **with arms swaying.***
- *The person is standing in place, **left leg kicking backward while left hand slapping it.***

Moreover, the overall motion description, such as *walking forward* or *standing in place* can be decomposed into the movement of individual body parts: $G = \{\text{Torso, Neck, Left Arm, Right Arm, Left Leg, Right Leg}\}$, as illustrated in Tab. 9.

Each body part $g \in G$ corresponds to a group of kinematic joints, as follows:

- **Torso:** Pelvis, spine joints (1–3 for SMPL [10]²)
- **Neck:** Neck, Head, Left/Right Collar
- **Left Arm:** Left Shoulder, Left Elbow, Left Wrist
- **Right Arm:** Right Shoulder, Right Elbow, Right Wrist
- **Left Leg:** Left Hip, Left Knee, Left Ankle
- **Right Leg:** Right Hip, Right Knee, Right Ankle

²<https://files.is.tue.mpg.de/black/talks/SMPL-made-simple-FAQs.pdf>

This new formulation is not only kinematically reasonable but also aligns seamlessly with natural language descriptions.

For each group $g \in G$ of joints at time t , we define:

1. *Group Position*: $\mathbf{P}_g(t) = \frac{1}{|J_g|} \sum_{j \in J_g} \mathbf{p}_j(t)$
2. *Limb Angles*: $\Theta_g(t) = \{\mathbf{R}_j(t) \mid j \in J_g\}$
3. *Group Velocity*: $\mathbf{V}_g(t) = \frac{1}{|J_g|} \sum_{j \in J_g} \mathbf{v}_j(t)$

We define the relationships between each pair $(g, h) \in G \times G$ of kinematic groups as:

1. *Relative Position*: $\Delta \mathbf{P}_{g,h}(t) = \mathbf{P}_h(t) - \mathbf{P}_g(t)$
2. *Relative Limb Angles* (angles of the connecting joint between two physically connected groups): $\Delta \Theta_{g,h}(t) = \Theta_{h \cap g}(t)$
3. *Relative Velocity* (angular velocity of the connecting joint between two physically connected groups): $\Delta \mathbf{V}_{g,h}(t) = \mathbf{V}_h(t) - \mathbf{V}_g(t)$

The Joint-Group Formulation is then defined as the collection of all these functions over time:

$$\begin{aligned} \mathcal{M}_{\text{group}} = \{ & (\mathbf{p}_j(g), \Theta_j(g), \mathbf{v}_j(g)) \mid g \in G, t \in T \} \\ & \cup \{ (\Delta \mathbf{P}_{g,h}(t), \Delta \Theta_{g,h}(t), \Delta \mathbf{V}_{g,h}(t)) \mid g \in G, t \in T \} \\ & \cup \{ \boldsymbol{\omega}_{\text{root}}(t), \mathbf{v}_{\text{root}}(t), h_{\text{root}}(t) \mid t \in T \} \\ & \cup \{ c_j(t) \mid j \in J_{\text{foot}}, t \in T \}. \end{aligned}$$

As demonstrated, the joint-level formulation can be transformed into the group-level formulation by aggregating joint data within each kinematic group. However, the reverse transformation is more challenging. To explore this reverse transformation, the following proposition holds:

Proposition 1. *Under the assumption that each kinematic group $g \in G$ moves as a rigid body, meaning the internal joint configurations within the group remain constant over time, the data of Keypoint-Level Formulation \mathcal{M} can be reconstructed from the data of Joint-Group Formulation $\mathcal{M}_{\text{group}}$*

Proof. Under the assumption that each kinematic group $g \in G$ moves as a rigid body, we have

- For all $j \in J_g$, the local position \mathbf{p}_j^g of joint j in the group's coordinate system is constant: $\mathbf{p}_j^g = \text{constant}$
- The group moves as a rigid body with rotation $\mathbf{R}_g(t)$ and translation $\mathbf{P}_g(t)$.

The objective is to reconstruct the joint positions $\mathbf{p}_j(t)$, joint angle $\mathbf{R}_j(t)$ and velocities $\mathbf{v}_j(t)$ for all $j \in J$.

Part A: Reconstruction of Joint Positions.

For each joint $j \in J_g$, the global position $\mathbf{p}_j(t)$ is given by:

$$\mathbf{p}_j(t) = \mathbf{R}_g(t) \mathbf{p}_j^g + \mathbf{P}_g(t) \quad (4)$$

where $\mathbf{R}_g(t)$ rotates the constant local joint position \mathbf{p}_j^g into the global coordinate system, and $\mathbf{P}_g(t)$ translates the rotated position to the global position. Since \mathbf{p}_j^g is constant

and known, and $\mathbf{R}_g(t)$ and $\mathbf{P}_g(t)$ are given from the group-level data, $\mathbf{p}_j(t)$ can be directly computed.

Part B: Reconstruction of Joint Angles.

Since we make no changes to the angles part, $\{\Delta \Theta_{g,h}(t) \mid g \in G\} \cup \{\Delta \Theta_{g,h}(t) \mid g, h \in G\} = \{\mathbf{R}_j(t) \mid j \in J\}$ for $t \in T$. So, the joint angle $\mathbf{R}_j(t)$ can be derived from the Joint-Group Formulation by arrangement.

Part C: Reconstruction of Joint Velocities.

First, we compute the time derivative of $\mathbf{p}_j(t)$ as:

$$\mathbf{v}_j(t) = \frac{d\mathbf{p}_j(t)}{dt} = \frac{d}{dt} (\mathbf{R}_g(t) \mathbf{p}_j^g + \mathbf{P}_g(t)) \quad (5)$$

Since \mathbf{p}_j^g is constant:

$$\mathbf{v}_j(t) = \left(\frac{d\mathbf{R}_g(t)}{dt} \right) \mathbf{p}_j^g + \frac{d\mathbf{P}_g(t)}{dt} \quad (6)$$

Recall that:

$$\frac{d\mathbf{R}_g(t)}{dt} = \Delta \hat{\mathbf{V}}_g(t) \mathbf{R}_g(t), \quad (7)$$

where $\Delta \hat{\mathbf{V}}_g(t)$ is the skew-symmetric matrix corresponding to $\Delta \mathbf{V}_g(t)$. Therefore:

$$\left(\frac{d\mathbf{R}_g(t)}{dt} \right) \mathbf{p}_j^g = \Delta \hat{\mathbf{V}}_g(t) \mathbf{R}_g(t) \mathbf{p}_j^g \quad (8)$$

$$\mathbf{v}_j(t) = \Delta \hat{\mathbf{V}}_g(t) \mathbf{R}_g(t) \mathbf{p}_j^g + \mathbf{V}_g(t) \quad (9)$$

Since:

$$\mathbf{p}_j(t) - \mathbf{P}_g(t) = \mathbf{R}_g(t) \mathbf{p}_j^g \quad (10)$$

We can rewrite:

$$\mathbf{v}_j(t) = \Delta \hat{\mathbf{V}}_g(t) (\mathbf{p}_j(t) - \mathbf{P}_g(t)) + \mathbf{V}_g(t) \quad (11)$$

The term $\mathbf{p}_j(t) - \mathbf{P}_g(t)$ represents the position of joint j relative to the group's center in global coordinates.

When combining Parts A, B and C, under the Rigid Body Assumption (i.e., each kinematic group moves as a rigid body), the joint-level formulation \mathcal{M} can be reconstructed from the group-level formulation $\mathcal{M}_{\text{group}}$, subject to a rigid transformation within each group. Consequently, identifying the natural language description of $\mathcal{M}_{\text{group}}$ uniquely corresponds to the joint-level formulation of the motion \mathcal{M} . However, this reconstruction is valid only under the rigid body assumption. In many human motions, the joints within a group (e.g., elbow bending within the arm group) can approximate rigid motion in specific cases, such as waving arms, swaying arms, or running, where limb angles remain relatively constant during a single motion. To address this limitation, we segment the motion temporally based on the keyframe, as outlined in Section B, ensuring that the rigid body assumption approximately holds within each motion fragment. \square

System Prompt

You are a motion description labeler who should describe the motion using your language as detailed as possible. Now, describe the motion in the given video or text by describing the motion of each kinematic group respectively: Kinematic groups: torso, neck, left arm, right arm, left leg, right leg.

To describe a motion, you should describe Inside each group and Between groups:

INDIVIDUAL GROUP: torso: <MOTION DESCRIPTION>; neck: <MOTION DESCRIPTION>; left arm: <MOTION DESCRIPTION>; right arm: <MOTION DESCRIPTION>; left leg: <MOTION DESCRIPTION>; right leg: <MOTION DESCRIPTION>;

BETWEEN GROUPS: torso AND neck: <MOTION DESCRIPTION>; torso AND left arm: <MOTION DESCRIPTION>; torso AND right arm: <MOTION DESCRIPTION>; torso AND left leg: <MOTION DESCRIPTION>; torso AND right leg: <MOTION DESCRIPTION>; neck AND left arm: <MOTION DESCRIPTION>; neck AND right arm: <MOTION DESCRIPTION>; neck AND left leg: <MOTION DESCRIPTION>; neck AND right leg: <MOTION DESCRIPTION>; left arm AND right arm: <MOTION DESCRIPTION>; left arm AND left leg: <MOTION DESCRIPTION>; left arm AND right leg: <MOTION DESCRIPTION>; right arm AND left leg: <MOTION DESCRIPTION>; right arm AND right leg: <MOTION DESCRIPTION>; left leg AND right leg: <MOTION DESCRIPTION>;

For each <MOTION DESCRIPTION>, you should describe the motion using language from the following perspective: Position (move from a place to a place. For example, the right hand goes through a rotation, moving primarily from a down position to extend horizontally), Axis-angle (How much degree the limb is bent and How the bending kinematic group is moving or rotating. For example, the right arm bends at the elbow to about 90 degrees while reaching outwards)

In your description, you can use simple adjectives or numerical scale (distance/degree/speed) to describe each motion (for example, push forward for 3 meters, from 0 to 45 degrees, etc).

Also, as the motion may change over time, you should consider the pose change at different timeframe. For example, the left arm first slap the left leg, then left arm hold high. Your description should also cover the pose variance over time.

Think through this part and infer the motion description based on the pose description given below

Based on the above formulation, write the description for the motion given by the user. You will also be given the pose descriptions of key frames. The keyframes can be used as reference and constraint, but don't mention the keyframes explicitly in your description, just make your description natural and casual.

User Prompt

Think about the motion: [a man stumbles to his right. the motion seems surprised so he was probably pushed. a person standing loses balance falling to the right and recovers standing. a person walks to the left. a person stumbles to the right and recovers their balance.], and constrain your motion description based on the given pose descriptions and image of key frames.

keyframes order: ['7', '22', '38', '60'] (Note that the fps of each motion will be 30. So you can infer the timing of each motion change based on the keyframe number, which can assist your description. For example, the person walks in place, then walk forward after 2 seconds. In this case, use time unit instead of keyframe number.)

Body pose descriptions of key frames: keyframe[7]:Their knees are straight, their elbows are bent a bit while their torso and both legs are straightened up while their left hand is past shoulder width apart from the other while their feet and their knees are about shoulder width apart.; keyframe[22]:Both hands are spread apart while their right calf and their torso are upright with their left knee slightly bent, located in front of their right knee with their right elbow nearly bent, their left elbow is rather bent with their knees and both feet shoulder width apart and their right knee straight.; keyframe[38]:Their torso is straightened up with their knees and their right elbow bent a bit. Their right hand is behind their back and located behind their left hand and their hands are apart wider than shoulder width while their knees are separated at shoulder width while their left elbow is rather bent.; keyframe[60]:The feet are approximately shoulder width apart, both hands are apart wider than shoulder width while both elbows are nearly bent, the left knee is shoulder width apart from the right knee while the left calf, the torso and the right leg are vertical, both knees are unbent.

image of key frames: <If presented, following by the key frames order, a sequence of images ignored here for efficiency>

Table 8. Example Prompt to Query GPT-4o for Automatic Dataset Annotation.

Global Action (from the original HumanML3D):

a man kicks something or someone with his left leg.#a/DET man/NOUN kick/VERB something/PRON or/CCONJ someone/PRON with/ADP his/DET left/ADJ leg/NOUN#0.0#0.0 the standing person kicks with their left foot before going back to their original stance.#the/DET stand/VERB person/NOUN kick/VERB with/ADP their/DET left/ADJ foot/NOUN before/ADP go/VERB back/ADV to/ADP their/DET original/ADJ stance/NOUN#0.0#0.0

a man kicks with something or someone with his left leg.#a/DET man/NOUN kick/VERB with/ADP something/PRON or/CCONJ someone/PRON with/ADP his/DET left/ADJ leg/NOUN#0.0#0.0

he is flying kick with his left leg#he/PRON is/AUX fly/VERB kick/NOUN with/ADP his/DET left/ADJ leg/NOUN#0.0#0.0

Kinematic Group:

Torso: The torso remains relatively upright throughout the motion, first positioned straight at the beginning, then slightly bending towards the left during the kick, and returning to an upright stance after the kick is executed.

Neck: The neck maintains stability, tilting slightly forward as the torso shifts, aligning with the direction of the kick. It helps maintain balance without significant rotation during the kick action.

Left Arm: The left arm initially starts relatively low near the hip before swinging upward and outward during the kick, creating momentum. It returns close to the body after the kick, maintaining a slight bend at the elbow.

Right Arm: The right arm extends and reaches upward, almost parallel to the ground, creating a counterbalance for the action. It remains relatively still after the kick, maintaining a slight bend at the elbow.

Left Leg: The left leg is the primary focus, starting from a straightened position at shoulder width. It then swings forward, bending at the knee to execute the kick, ultimately extended fully forward before returning to the original stance.

Right Leg: The right leg remains grounded throughout the motion, initially straightening and slightly shifting back for balance while the left leg is in action, then returning to its original position once the kick is completed.

Group Interaction:

Torso AND Neck: The torso and neck move harmoniously; as the torso bends slightly left during the kick, the neck tilts forward in alignment with the action, maintaining a strong posture while providing support.

Torso AND Left Arm: As the torso initiates the kick, the left arm swings backward to create momentum, moving in a coordinated manner before returning closer to the torso after the kick is completed.

Torso AND Right Arm: The right arm serves as a counterbalance; as the torso shifts slightly to the left, the right arm reaches higher, creating stability and balance during the kick.

Torso AND Left Leg: The left leg extends forward as the torso maintains its upright position, the torso adjusts to the kick's momentum but quickly stabilizes once the leg retracts.

Torso AND Right Leg: The right leg supports the body weight throughout the kick, keeping the torso aligned; as the left leg kicks, the right leg remains slightly bent at the knee for stability.

Neck AND Left Arm: As the left arm swings upward to support the kick, the neck aligns to follow the movement, enhancing the overall motion without excessive twisting.

Neck AND Right Arm: The right arm's upward reach aids in stabilizing the movement, and the neck maintains a forward position while providing visual direction during the kick.

Neck AND Left Leg: The left leg moves forward while the neck follows the general direction of the kick, keeping a low tilt as it helps maintain focus on the target.

Neck AND Right Leg: The right leg remains grounded as the neck tilts very slightly forward without excessive rotation while observing the left leg's action. left arm AND right arm: The left arm moves upward for a moment as the right arm extends higher, providing balance; both arms slightly bend at the elbows during the action.

Left Arm AND Left Leg: The left arm and left leg perform synchronous movement; the arm swings to aid the momentum of the extended leg during the kick phase and returns together afterward.

Left Arm AND Right Leg: The left arm raises slightly while the right leg remains positioned on the ground to help maintain balance; they coordinate as the kick progresses.

Right Arm AND Left Leg: The right arm reaches out while the left leg is kicked forward, the actions working in tandem to support the balance during the execution.

Right Arm AND Right Leg: The right arm extends higher as the right leg stays grounded, providing support without conflicting with each other. left leg AND right leg: The left leg moves forward as the right leg remains stable on the ground, creating a contrast in movement, with the left leg bending as it prepares for the kick and extending forward while the right leg holds strong.

Table 9. Example Motion Descriptions in Our Augmented HumanML3D Dataset.

References

- [1] Wenxun Dai, Ling-Hao Chen, Jingbo Wang, Jinpeng Liu, Bo Dai, and Yansong Tang. MotionLCM: Real-time Controllable Motion Generation via Latent Consistency Model. *arXiv preprint arXiv:2404.19759*, 2024. 5, 6
- [2] Ginger Delmas, Philippe Weinzaepfel, Thomas Lucas, Francesc Moreno-Noguer, and Grégory Rogez. Posescript: 3D human poses from natural language. In *European Conference on Computer Vision*, pages 346–362, 2022. 1
- [3] Chuan Guo, Xinxin Zuo, Sen Wang, Shihao Zou, Qingyao Sun, Annan Deng, Minglun Gong, and Li Cheng. Action2motion: Conditioned generation of 3D human motions. In *ACM International Conference on Multimedia*, pages 2021–2029, 2020. 5
- [4] 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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5152–5161, 2022. 3, 5, 6, 7
- [5] Chuan Guo, Xinxin Zuo, Sen Wang, and Li Cheng. TM2T: Stochastic and Tokenized Modeling for the Reciprocal Generation of 3D Human Motions and Texts. In *European Conference on Computer Vision*, 2022. 5
- [6] Chuan Guo, Yuxuan Mu, Muhammad Gohar Javed, Sen Wang, and Li Cheng. MoMask: Generative masked modeling of 3D human motions. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. 2, 4, 5, 6
- [7] Pinxin Liu, Luchuan Song, Junhua Huang, and Chenliang Xu. GestureLSM: Latent Shortcut based Co-Speech Gesture Generation with Spatial-Temporal Modeling. *arXiv preprint arXiv:2501.18898*, 2025. 5
- [8] Xinpeng Liu, Yong-Lu Li, Ailing Zeng, Zizheng Zhou, Yang You, and Cewu Lu. Bridging the gap between human motion and action semantics via kinematic phrases. *arXiv preprint arXiv:2310.04189*, 2023. 7
- [9] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv preprint arXiv:1907.11692*, 2019. 2
- [10] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. Smpl: A skinned multi-person linear model. In *Seminal Graphics Papers: Pushing the Boundaries, Volume 2*, pages 851–866, 2023. 7
- [11] OpenAI. GPT-4 Technical Report. *arXiv preprint arXiv:2303.08774*, 2024. 1, 2, 5
- [12] 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. 2
- [13] Mathis Petrovich, Michael J Black, and Gül Varol. TMR: Text-to-motion retrieval using contrastive 3D human motion synthesis. In *IEEE/CVF International Conference on Computer Vision*, pages 9488–9497, 2023. 2
- [14] Mathis Petrovich, Or Litany, Umar Iqbal, Michael J. Black, Gül Varol, Xue Bin Peng, and Davis Rempe. Multi-Track Timeline Control for Text-Driven 3D Human Motion Generation. In *Workshop on Human Motion Generation, IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. 5
- [15] Ekkasit Pinyoanuntapong, Pu Wang, Minwoo Lee, and Chen Chen. MMM: Generative masked motion model. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1546–1555, 2024. 4, 5, 6
- [16] Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. *arXiv preprint arXiv:1908.10084*, 2019. 1
- [17] Yoni Shafir, Guy Tevet, Roy Kapon, and Amit Haim Bermano. Human Motion Diffusion as a Generative Prior. In *International Conference on Learning Representations*, 2024. 6
- [18] 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. 6
- [19] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, 2017. 2
- [20] Weilin Wan, Zhiyang Dou, Taku Komura, Wenping Wang, Dinesh Jayaraman, and Lingjie Liu. TLControl: Trajectory and Language Control for Human Motion Synthesis. *arXiv preprint arXiv:2311.17135*, 2024. 4
- [21] Yiming Xie, Varun Jampani, Lei Zhong, Deqing Sun, and Huaizu Jiang. OmniControl: Control Any Joint at Any Time for Human Motion Generation. In *International Conference on Learning Representations*, 2024. 4, 5, 6