ENCODER-FREE HUMAN MOTION UNDERSTANDING VIA STRUCTURED MOTION DESCRIPTIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

The world knowledge and reasoning capabilities of text-based large language models (LLMs) are advancing rapidly, yet current approaches to human motion understanding, including motion question answering and captioning, have not fully exploited these capabilities. Existing LLM-based methods typically learn motion-language alignment through dedicated encoders that project motion features into the LLM's embedding space, remaining constrained by cross-modal representation and alignment. Inspired by biomechanical analysis, where joint angles and body-part kinematics have long served as a precise descriptive language for human movement, we propose **Structured Motion Description (SMD)**, a rule-based, deterministic approach that converts joint position sequences into structured natural language descriptions of joint angles, body part movements, and global trajectory. By representing motion as text, SMD enables LLMs to apply their pretrained knowledge of body parts, spatial directions, and movement semantics directly to motion reasoning, without requiring learned encoders or alignment modules. We show that this approach goes beyond state-of-the-art results on both motion question answering (66.7% on BABEL-QA, 90.1% on HuMMan-QA) and motion captioning (R@1 of 0.584, CIDEr of 53.16 on HumanML3D), surpassing all prior methods. SMD additionally offers practical benefits: the same text input works across different LLMs with only lightweight LoRA adaptation (validated on 8 LLMs from 6 model families), and its human-readable representation enables interpretable attention analysis over motion descriptions. The code is available in Supplementary Materials.

1 INTRODUCTION

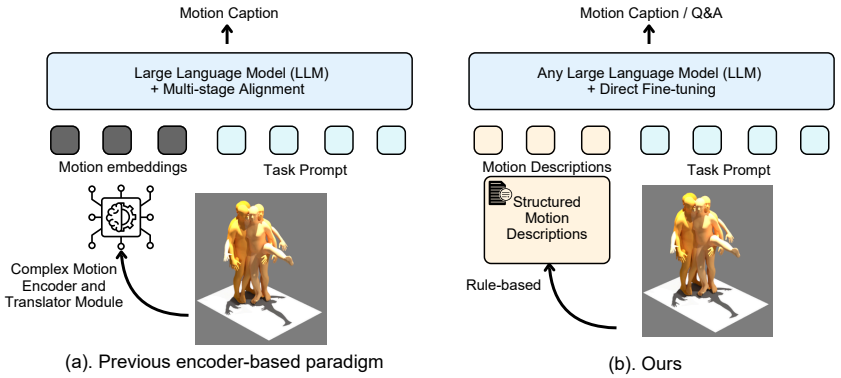


Figure 1: Comparison of our encoder-free approach (b) against prior encoder-based methods (a).

Understanding human motion from skeletal data, typically represented as sequences of 2D or 3D body joints connected by bones over time, is a fundamental problem in computer vision, underpinning tasks such as motion question answering (QA) and motion captioning. Motion QA (Endo et al., 2023; Li et al., 2025a) entails answering questions about a given motion sequence, such as which body part is moving, in which direction, or what action is being performed, which requires fine-grained

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

spatiotemporal reasoning over joint-level motion details. Motion captioning (Guo et al., 2022a; Zhu et al., 2025) requires generating a natural language description that summarizes the overall motion content, bridging the gap between low-level skeletal data and high-level semantic understanding. The rapid advancement of large language models (LLMs) has motivated recent efforts to tackle these tasks with LLMs, leveraging their world knowledge and reasoning capabilities for natural language interaction with motion data.

Most of these LLM-based approaches (Zhu et al., 2025; Wu et al., 2025; Jiang et al., 2023; Chen et al., 2025b; Wang et al., 2026) follow an encoder-based paradigm: a learned motion encoder projects sequences of joint positions into the LLM's token embedding space, using either discrete tokenization via VQ-VAE (Jiang et al., 2023; Wu et al., 2025), continuous latent encoding via VAE (Zhu et al., 2025), or linear projection (Chen et al., 2025b), following the multi-modal alignment framework established in vision-language models (VLMs) (Liu et al., 2023). These methods have driven substantial progress, yet the reliance on cross-modal representation and alignment introduces practical considerations: the motion encoder and alignment module are typically trained through multi-stage pipelines with paired motion-text data, the resulting system is coupled to a specific LLM backbone, and the learned motion tokens are not human-readable.

As illustrated in Fig. 1, we take an encoder-free approach by introducing Structured Motion Description (SMD): a deterministic, rule-based conversion from joint-position sequences into structured natural-language text, which is then fed directly to LLMs. This idea leverages the fact that LLMs, pretrained on vast corpora, already encode extensive knowledge of body parts, spatial directions, and movement semantics (Li et al., 2025b). By replacing learned motion embeddings with descriptive text, SMD enables LLMs to process motion in their native modality.

The key challenge is to produce descriptions that are precise enough for fine-grained reasoning while using concepts the LLM already understands. Biomechanics provides a natural foundation: joint angles and body-part kinematics have long served as a precise descriptive language for human movement (Perry & Burnfield, 2024; Chen et al., 2025a; Zhang et al., 2026). Clinical gait analysis, for example, characterizes walking via time-varying flexion curves of the hip, knee, and ankle. These descriptors are inherently textual; a statement like "hip flexion increases from 3° to 81°" is simultaneously a quantitative measurement and a natural-language sentence. SMD extends this idea to full-body motion by computing angular changes across all major joints together with global trajectory information.

Because SMD uses natural language terms for body parts, spatial directions, and temporal patterns, it naturally leverages the LLM's world knowledge of these concepts (Li et al., 2025b), as validated by our zero-shot experiments in Section 4. With lightweight LoRA (Hu et al., 2021) fine-tuning as the only training step, our approach delivers three key advantages: (1) **Strong performance with minimal training**: SMD surpasses prior state-of-the-art on motion QA (Endo et al., 2023; Li et al., 2025a) and captioning (Guo et al., 2022a) without any learned motion encoder or multimodal alignment module, outperforming both dedicated motion models and encoder-based LLM approaches; (2) **LLM-agnostic flexibility at low cost**: the same text input works across different LLMs by attaching a lightweight LoRA adapter ($\sim$40M parameters, 2–8 GPU-hours on a single H200), without retraining a motion encoder or alignment module; we validate this across 8 LLMs from 6 model families; (3) **Built-in interpretability**: attention analysis over the human-readable SMD tokens directly reveals which body parts and trajectory segments the model relies on for generation, offering transparent, fine-grained interpretability by design.

2 Related Work

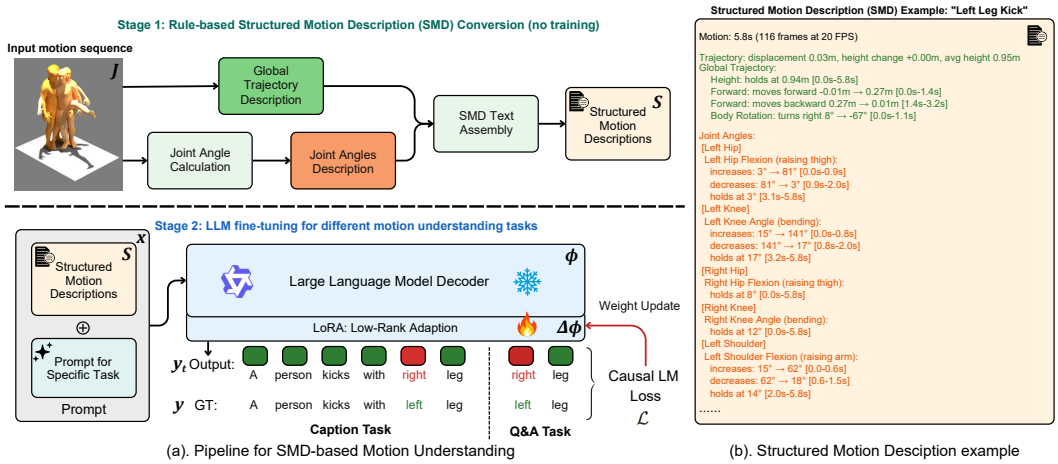
Motion representation. 3D human motion is most commonly represented as joint-position sequences from parametric body models such as SMPL (Loper et al., 2023), typically via the 263-dim HumanML3D feature (Guo et al., 2022a) (root velocities, local joint positions, 6D joint rotations (Zhou et al., 2019), joint velocities, and foot contact labels); while expressive, these high-dimensional numerical features require learned encoders to interface with LLMs. A complementary line uses interpretable, human-readable descriptors: PoseScript (Delmas et al., 2024) and PoseFix (Delmas et al., 2023) generate rule-based natural-language descriptions of static poses based on joint positions and predefined rules for retrieval and correction tasks. LaMoGen (Jiang et al., 2026) map motion to a Laban-inspired symbolic alphabet paired with a trained codebook and decoder for

motion generation. Zhang et al. (2026) map biomechanical joint angles from skeleton sequences into structured pseudo-images for motion retrieval with vision transformers. Our work also builds on joint-angle representations, but converts them into structured text rather than images, enabling any LLM to process motion directly without learned visual encoders.

Motion understanding. Human motion understanding encompasses a range of tasks that reason over skeletal sequences. We focus on two representative tasks: motion QA and motion captioning. For motion QA, early methods use task-specific architectures without pretrained LLMs: NSPose (Endo et al., 2023) introduces the BABEL-QA benchmark with a neuro-symbolic framework that parses questions into compositional programs executed by hand-designed modules over learned motion features, while IMoRe (Li et al., 2025a) replaces these modules with implicit program-guided reasoning. MotionLLM (Chen et al., 2025b) was the first to apply a billion-parameter LLM (Vicuna-13B) via a learned motion encoder, achieving performance comparable to IMoRe. HuMoCon (Fang et al., 2025) and FiGMo (Markhorst et al., 2026) collect large amounts of motion and video data to train the LLM to understand motion tokens, whereas we convert motion into text and directly leverage the LLM's world knowledge.

For motion captioning, early approaches build on motion-text joint embedding spaces (TM2T (Guo et al., 2022b), LaMP (Li et al., 2025c), MoTe (Wu et al., 2024)). With the rise of LLMs, recent methods adopt the encoder-alignment paradigm of VLMs such as LLaVA (Liu et al., 2023): MotionGPT (Jiang et al., 2023), MotionGPT-2 (Wang et al., 2026), MoChat (Mo et al., 2024), MotionGPT3 (Zhu et al., 2025), and MG-MotionLLM (Wu et al., 2025) all pair a learned motion encoder (VQ-VAE or VAE tokens) with an LLM backbone. GeoMotionGPT (Ye et al., 2026) aligns motion VAE tokens orthogonally to the LLM space but requires full LLM retraining. Li et al. (2025b) find that pretrained LLMs already possess knowledge of body parts and physics but need adaptation for precise motion tasks. Beyond motion, text-mediated LLM inputs have been explored in video understanding by works such as LLoVi (Zhang et al., 2024) and Socratic Models (Zeng et al., 2022), but these rely on a learned caption model (e.g., a VLM) to produce the text, inheriting its biases and losing fine-grained spatial-temporal detail. Skeletal sequences instead have a structured, low-dimensional form that admits precise, deterministic conversion to text without any learned component. Our work converts motion into structured biomechanical descriptions, enabling any LLM to perform motion understanding without learned encoders or cross-modal alignment.

3 METHOD



(a). Pipeline for SMD-based Motion Understanding (b). Structured Motion Desciption example

Figure 2: Overview of our approach.(a) Stage 1: conversion of skeleton sequences into SMD; Stage 2: SMD fed into an LLM fine-tuned with LoRA. (b) Truncated SMD for a "Left Leg Kick" motion, organized into a meta-information header, a global trajectory block, and a joint angles block grouped by body part. Only the three most active joints are shown; the full 26-angle SMD is in Appendix A.1.

As illustrated in Figure 2, our approach consists of two stages: (1) a rule-based conversion f_{SMD} from joint positions to an SMD, and (2) LoRA fine-tuning of a pretrained LLM that takes this text as input. No learned motion encoder, VQ-VAE, or cross-modal alignment module is involved.

3.1 Structured Motion Description

$f_{\text{SMD}} : \mathbb{R}^{T\times J\times 3} \to \mathcal{V}^*$ is a deterministic function that maps a sequence of joint positions $\mathbf{J} = [\mathbf{j}_1, \ldots, \mathbf{j}_T] \in \mathbb{R}^{T\times J\times 3}$ (where $J = 22$ for SMPL-based representation and T is the number of frames) to a text string $\mathcal{S} \in \mathcal{V}^*$ over the LLM's vocabulary $\mathcal{V}$. As illustrated in Figure 2(a), the conversion proceeds in the following four steps.

Step 1: Joint angle calculation. The goal of this step is to produce a set of K joint-angle channels $\{\theta_k^{(1:T)}\}_{k=1}^K$, each a scalar time series of length T that describes the angle at one anatomical joint over the input frames. Following standard biomechanical conventions (Zhang et al., 2026; Schlegel et al., 2024; Wu et al., 2002; 2005), we define these angles by projecting bone vectors onto anatomical reference planes within joint-local coordinate frames, organized along a kinematic chain rooted at a body-local coordinate frame.

The body-local coordinate frame is a right-handed reference system attached to the pelvis, with $\mathbf{e}_x$ pointing forward, $\mathbf{e}_y$ pointing up, and $\mathbf{e}_z$ pointing laterally (right). It defines the subject's overall facing direction and serves as the root of the kinematic chain. At each time frame t, we construct it from three landmark joints (pelvis, left hip, right hip), and then compute joint angles sequentially along the chain. For example, hip flexion is measured in the sagittal ($\mathbf{e}_x$–$\mathbf{e}_y$) plane as the signed angle of the femur vector $\mathbf{v}_{\text{fem}} = \mathbf{j}_{\text{knee}} - \mathbf{j}_{\text{hip}}$ from the downward direction $-\mathbf{e}_y$ toward $\mathbf{e}_x$:

$$\theta_{\text{hip-flex}}^{(t)} = \operatorname{atan2}\Big(\hat{\mathbf{v}}_{\text{fem}}^{(t)} \cdot \mathbf{e}_x,\ -\hat{\mathbf{v}}_{\text{fem}}^{(t)} \cdot \mathbf{e}_y\Big). \tag{1}$$

This yields a signed angle in [-180°, 180°], so flexion (positive) and extension (negative) are distinguished naturally. Adduction is computed as the angle between the femur vector and its projection onto the sagittal plane, and rotation about the femur is computed using the tibia direction as a secondary reference. Other angles, such as knee flexion and shoulder adduction, follow analogous definitions using appropriate bone vectors and reference planes (Appendix A.1 for the complete list).

In total, this procedure yields $K = 26$ biomechanical joint angles $\boldsymbol{\theta}^{(t)} = [\theta_1^{(t)}, \ldots, \theta_K^{(t)}]$, organized into 13 body-part groups spanning the pelvis, lumbar spine, neck, bilateral hip, knee, ankle, shoulder, and elbow. Because these angles follow standard anatomical conventions rather than skeleton-specific indices, SMD generalizes to other skeleton formats (e.g., Kinect, COCO) provided the relevant landmarks are available, as we demonstrate on NTU RGB+D 60 in Appendix C.1; each angle also carries a clear anatomical meaning (e.g., hip flexion = raising the thigh, shoulder adduction = moving the arm inward), enabling a direct translation into text.

Step 2: Global trajectory description. Joint angles describe how body parts are posed relative to one another, but they are invariant to the subject's global position and orientation in the world. Many motions (e.g., walking, jumping, turning), however, are also defined by global displacement of the body through space. We characterize global motion using the pelvis position, since it is the root joint of the SMPL skeleton. The pelvis also closely approximates the body's center of mass, making it a standard landmark for global motion analysis in biomechanics and gait research (Perry & Burnfield, 2024).

Specifically, we extract the pelvis trajectory along three translational axes (forward/backward, lateral, height) and one rotational axis (body yaw, computed from the forward direction $\mathbf{e}_x$ established in Step 1). Each axis is segmented into contiguous intervals using a two-stage procedure: the time series is first smoothed with a moving average filter of window size $w = 7$ frames (0.35s at 20 FPS, short enough to preserve rapid voluntary movements while filtering frame-level noise), and then partitioned by peak-valley detection whenever the change between consecutive extrema exceeds a minimum threshold τ_p (0.03m for translation, to ignore positional jitter; 15° for yaw, to distinguish deliberate turns from postural sway). Each resulting segment is classified according to the sign and axis of motion, yielding descriptions such as "Forward Position: moves forward 0.00m → 1.23m [0.0s–2.5s]" or "Body Rotation: turns left 0° → 45° [1.0s–1.8s]" (the complete set of descriptive phrases is listed

in Table 7, Appendix A.1). We denote the resulting set of trajectory segments as $\{s_{\text{traj}}\}$. A sensitivity analysis of the Segmentation thresholds τ_p and Smoothing window w is provided in Table 4.

Step 3: Joint angles description. We apply the same smoothing and peak-valley segmentation procedure as in Step 2 to each of the K time series $\theta_k^{(1:T)}$ from Step 1, using a minimum angular change threshold of $\delta = 5°$ (set above the noise floor of the input data to avoid spurious segments; sensitivity analysis in Table 4). Each segment is labeled with one of four types based on the boundary values v_{start} and v_{end} of the segment: *increases* if the angle rises by at least δ; *decreases* if it falls by at least δ; *holds* if the change is smaller than δ; and *repeats N cycles* if consecutive moving segments alternate in direction with consistent amplitude (Appendix A.1), capturing periodic patterns such as cyclic hip and knee flexion during walking.

As a concrete example, a left hip flexion angle that rises from 3° to 81° over the first 18 frames, falls back to 7° by frame 40, and then stays near 3° is represented as three segments: "Left Hip Flexion (raising thigh): *increases* 3° → 81° [0.0s–0.9s], *decreases* 81° → 7° [0.9s–2.0s], *holds at* 3° [2.0s–5.8s]". This compression reduces a raw time series of T values (e.g., 200 frames) into typically 3–8 descriptive intervals per joint, substantially shortening the text while preserving the temporal structure. We denote the resulting text segments as $\{s_k\}_{k=1}^{K}$, one group per joint angle.

Step 4: SMD assembly. The trajectory and joint angle segments from Steps 2 and 3 are assembled into a single structured text string $\mathcal{S} = f_{\text{assemble}}(\{s_k\}_{k=1}^{K}, \{s_{\text{traj}}\})$. As shown in the "Left Leg Kick" example in Figure 2(b), $\mathcal{S}$ is organized hierarchically into three blocks:

- **Meta information**: a one-line header reporting motion duration, frame count, and frame rate (e.g., "Motion: 5.8s (116 frames at 20 FPS)").
- **Global trajectory**: an aggregated summary (overall displacement, height change, average height) followed by per-axis trajectory segments from Step 2 (e.g., "Forward: moves forward -0.01m → 0.27m [0.0s–1.4s]"; "Body Rotation: turns right 8° → -67° [0.0s–1.1s]").
- **Joint angles**: per-joint angle segments from Step 3, grouped by body part with section headers (e.g., "[Left Hip]", "[Left Knee]").

Once tokenized by the LLM, $\mathcal{S}$ occupies approximately 4,000 tokens on HumanML3D for all 26 joint angles, or approximately 1,000 input tokens when only the Top-3 most active joints are selected.

3.2 Task Formulation and Training

We formulate both motion QA and motion captioning tasks as autoregressive text generation conditioned on the SMD. Let LLM_ϕ denote a pretrained LLM with frozen parameters ϕ. Given a motion $\mathbf{J}$, we compute $\mathcal{S} = f_{\text{SMD}}(\mathbf{J})$ and construct the text prompt x by concatenating a task-specific system instruction with $\mathcal{S}$; for QA we additionally append the question text q and the candidate answer options, and the target $y = [y_1, \ldots, y_L]$ is the text of the correct option, whereas for captioning the target y is a natural-language caption describing the motion. The multiple-choice QA format matches the standard protocol of existing motion QA benchmarks (Endo et al., 2023; Li et al., 2025a) and the evaluation setting of all baselines; it naturally extends to open-ended QA by omitting the options. The full prompt template for both tasks is shown in Appendix A.3.

During training, only the target response tokens y contribute to the loss; all tokens in x are masked. The training objective is the standard causal language modeling loss over y:

$$\mathcal{L} = -\sum_{l=1}^{L} \log p_{\phi+\Delta\phi}(y_l \mid \mathbf{x}, y_{1:l-1}), \tag{2}$$

where $\Delta\phi$ are LoRA (Hu et al., 2021) adapter parameters inserted into all linear layers, with the base LLM weights ϕ frozen. We train separate LoRA adapters for motion QA and motion captioning.

At inference, given a new motion $\mathbf{J}'$, we build the prompt $\mathbf{x}'$ from $\mathcal{S}' = f_{\text{SMD}}(\mathbf{J}')$ (without the target) and generate $\hat{\mathbf{y}} = \text{LLM}_{\phi+\Delta\phi}(\mathbf{x}')$ autoregressively; QA predictions are compared to the ground-truth answer, and captions are evaluated against references with standard metrics.

4 EXPERIMENTS AND ANALYSIS

4.1 EXPERIMENTAL SETUP

Datasets. We evaluate on BABEL-QA (Endo et al., 2023) (1,109 motions, 2,577 QA pairs, 1,800/384/393 train/val/test) and HuMMan-QA (Li et al., 2025a) (925 motions, 3,123 QA pairs, 2,066/524/533 train/val/test) for motion QA, and HumanML3D (Guo et al., 2022a) (14,616 motions with 44,970 captions) for motion captioning. Since the original QA benchmarks use variable option counts (4–20 for BABEL-QA, 6–155 for HuMMan-QA), making cross-method comparison difficult, we standardize both to a fixed 10-option format (details in Appendix D.1). All methods are retrained and evaluated on this format except MotionLLM (Chen et al., 2025b), whose training data and checkpoints are unavailable (a fair comparison with MotionLLM in Appendix B.1).

Evaluation metrics. For QA, we report accuracy via exact string matching following previous works (Chen et al., 2025b; Li et al., 2025a; Endo et al., 2023). For captioning, we use text-motion alignment metrics (R-Precision R@1/2/3 and MM-Distance computed with the pretrained T2M evaluator (Guo et al., 2022a)) and linguistic metrics (BLEU@1/4 (Papineni et al., 2002), ROUGE-L (Lin, 2004), CIDEr (Vedantam et al., 2015), BERTScore (Zhang et al., 2019)) following the common evaluation protocol used in prior works (Zhu et al., 2025; Wang et al., 2026; Wu et al., 2025; Jiang et al., 2023; Guo et al., 2022b). Since prior CIDEr computations inadvertently included extraneous symbols (e.g., colons, newlines) affecting scores, we re-evaluate CIDEr for all baselines on their released checkpoints with a cleaned implementation; details in Appendix D.2.

Implementation details. Our default backbone is Qwen2.5-7B-Instruct. LoRA (rank 16, $\alpha = 32$, dropout 0.05) is applied to all linear layers, yielding approximately 40M trainable parameters out of 7.6B total; the base model weights remain frozen. For QA, we train jointly on BABEL-QA (1,800 pairs) and HuMMan-QA (2,066 pairs) for 5 epochs with batch size 8, learning rate 1e-4, and cosine annealing (approximately 7 GPU-hours). For captioning, we train on HumanML3D train set (23,384 motions, each with 3–4 reference captions) for 5 epochs with batch size 8 and learning rate 1e-4 (approximately 20 GPU-hours). All experiments use a single NVIDIA H200 GPU.

Baselines. For QA, we compare against specialized motion QA models (NSPose (Endo et al., 2023), IMoRe (Li et al., 2025a)), an encoder-based motion LLM (MotionLLM (Chen et al., 2025b)), and MotionGPT3-Qwen. All QA baselines except MotionLLM (Chen et al., 2025b) are retrained on the standardized 10-option format for fair comparison. Both HuMoCon (Fang et al., 2025) and FiGMo (Markhorst et al., 2026) use an LLM to judge the output, since their code is not yet available, we didn't directly compare with them. For captioning, we compare against TM2T (Guo et al., 2022b), MotionGPT (Jiang et al., 2023), LaMP (Li et al., 2025c), MoTe (Wu et al., 2024), MotionGPT3 (Zhu et al., 2025), MG-MotionLLM (Wu et al., 2025), and MotionGPT3-Qwen. MotionGPT3-Qwen is a controlled baseline we construct by porting the MotionGPT3 paradigm (frozen VAE + MLP projection, LLaVA-style two-stage training) to our Qwen2.5-7B backbone, isolating the effect of motion representation while holding the LLM constant (construction details in Appendix D.1).

4.2 MAIN RESULTS

Motion QA. As listed in Table 1, our approach achieves 66.7% accuracy on BABEL-QA and 90.1% on HuMMan-QA, surpassing the previous state-of-the-art, IMoRe (Li et al., 2025a), by 6.6 and 14.9 percentage points, respectively. On BABEL-QA, MotionGPT3-Qwen, which uses the same Qwen2.5-7B backbone but replaces SMD with VAE-encoded latent tokens, achieves 50.1% accuracy, trailing our approach by 16.6 percentage points. On HuMMan-QA, the gap is larger: 22.0% for MotionGPT3-Qwen versus 90.1% for ours. This disparity likely stems from pretraining mismatch: the MotionGPT3 VAE was trained on HumanML3D, whereas HuMMan-QA motions were collected via RGB-D reconstruction. A domain-adaptation analysis (Appendix B.3) shows that fine-tuning the VAE on HuMMan recovers much of this gap ($22.0\% \rightarrow 63.2\%$). This indicates that encoder-based methods require per-domain fine-tuning of the encoder to handle cross-domain motion data, and even after adaptation, their accuracy remains below ours (63.2% vs. 90.1%). Our fully rule-based approach produces consistent representations regardless of the motion data source. Side-by-side qualitative QA examples are provided in Appendix E.1.

Table 1: Main results. Top: motion QA accuracy (%) on BABEL-QA and HuMMan-QA. Bottom: motion captioning performance on the HumanML3D test set. †Results from the original paper. ‡MotionGPT3 paradigm (pretrained VAE + MLP projection) replicated on the same Qwen2.5-7B backbone with LLaVA-style two-stage training.

MOTION QA				
Method	Input	Model	BABEL-QA	HuMMan-QA
NSPose (Endo et al., 2023)	joints	Specialized	48.1	70.9
IMoRe (Li et al., 2025a)	joints	Specialized	60.1	75.2
MotionLLM† (Chen et al., 2025b)	joints	Vicuna-13B	43.6	—
MotionGPT3-Qwen‡	joints	Qwen2.5-7B	50.1	22.0
Ours (SMD-based)	text	Qwen2.5-7B	**66.7**	**90.1**

MOTION CAPTIONING (M2T)									
Method	R@1↑	R@2↑	R@3↑	MMD↓	B@1↑	B@4↑	R-L↑	CIDEr↑	BS↑
TM2T (Guo et al., 2022b)	.516	—	.823	2.94	48.9	7.0	38.1	35.1	32.2
MotionGPT (Jiang et al., 2023)	.543	—	.827	2.82	48.2	12.5	37.4	39.4	32.4
LaMP (Li et al., 2025c)	.547	—	.831	2.81	47.8	13.0	37.1	39.7	32.7
MoTe (Wu et al., 2024)	.577	—	.871	2.65	46.7	11.2	37.4	41.3	30.3
MG-MotionLLM (Wu et al., 2025)	**.592**	—	.866	2.58	—	8.1	—	—	36.7
MotionGPT3 (Zhu et al., 2025)	.573	.773	.864	2.43	59.1	19.4	46.2	40.6	35.2
MotionGPT3-Qwen‡	.555	.753	.843	2.47	59.43	19.60	45.82	46.13	42.87
Ours (SMD-based)	.584	**.794**	**.883**	**2.35**	**63.45**	**22.67**	**47.80**	**53.16**	**45.58**

Table 2: Effect of the number of joints included in SMD. All experiments used Qwen2.5-7B.

Joints	Tokens	BABEL-QA	HuMMan-QA	R@1↑	R@2↑	R@3↑	MMD↓	B@1↑	B@4↑	R-L↑	CIDEr↑	BS↑
None	∼0.6K	56.2	67.4	.452	.626	.720	3.43	48.19	11.87	24.01	21.69	29.85
Top-3	∼1K	**73.3**	**91.0**	.547	.754	.842	2.59	61.74	22.25	47.52	52.28	44.94
Top-5	∼1.5K	68.2	90.8	.563	.757	.843	2.49	61.24	22.20	47.32	51.08	44.48
Top-10	∼2.2K	66.2	89.3	.569	.773	.860	2.44	61.66	22.41	47.45	51.82	44.97
Top-20	∼3.5K	69.2	89.1	.582	.793	.878	2.37	**63.48**	**23.07**	**48.30**	**56.03**	**46.22**
All-26	∼4K	66.7	90.1	**.584**	**.794**	**.883**	**2.35**	63.45	22.67	47.80	53.16	45.58

Motion captioning. On HumanML3D (Table 1), our SMD-based approach achieves the best results on all metrics except R@1. Against the strongest prior method MotionGPT3 (Zhu et al., 2025), we improve R-Precision at every level and MM-Distance, indicating tighter text-motion semantic alignment. The gains in text generation quality are also substantial: $+17\%$ relative on BLEU@4, $+31\%$ on CIDEr, showing higher consensus with multiple references and closer stylistic alignment to human annotations. BERTScore improves from 35.23 to 45.58, confirming stronger semantic similarity at the token embedding level. The same-backbone controlled baseline MotionGPT3-Qwen (Qwen2.5-7B) consistently trails our approach across all metrics, and scaling its projection MLP to 32/64/128 tokens (Appendix D.4) does not close the gap because the larger MLPs overfit on the limited training data. Side-by-side qualitative captioning examples are provided in Appendix E.2.

4.3 ABLATION STUDIES

Number of joints. Table 2 varies how many joints are included in SMD affects the performance."None" uses only global trajectory, "Top-K" selects the K joints with the largest angular displacement per motion, and "All-26" includes all joints. QA and captioning respond differently. For QA, Top-3 is already best (73.3%/91.0%), as focusing on the most active joints filters out noise from static joints. For captioning, retrieval metrics rise monotonically with more joints (R@1: 0.452 with trajectory-only $\rightarrow$ 0.584 with All-26), since open-ended generation benefits from richer full-body descriptions. The optimal SMD granularity is thus task-dependent.

Trajectory representation. Table 3 compares three ways to encode global trajectory: none, egocentric (body-relative directions), and absolute (world coordinates). Absolute trajectory achieves the best overall performance, particularly on retrieval metrics (R@1 = 0.584). Notably, removing trajectory entirely does not cause a dramatic drop: BABEL-QA accuracy decreases by only 1.8 points and captioning CIDEr remains comparable (53.34 vs. 53.16). This may suggest that the model can

Table 3: Effect of trajectory representation in the SMD (All-26 joints, Qwen2.5-7B).

Trajectory	BABEL-QA	HuMMan-QA	R@1↑	R@2↑	R@3↑	MMD↓	B@1↑	B@4↑	R-L↑	CIDEr↑	BS↑
None	64.9	**90.1**	.574	.778	.868	2.41	62.26	22.11	47.69	**53.34**	**45.62**
Egocentric	**66.9**	89.3	.558	.762	.852	2.53	62.20	21.60	47.17	50.09	44.94
Absolute	66.7	**90.1**	**.584**	**.794**	**.883**	**2.35**	**63.45**	**22.67**	**47.80**	53.16	45.58

Table 4: Sensitivity to SMD conversion parameters (All-26 joints, Qwen2.5-7B). The three sub-blocks varying one parameter while holding the others at their defaults.

Setting	BABEL-QA	HuMMan-QA	R@1↑	R@2↑	R@3↑	MMD↓	B@1↑	B@4↑	R-L↑	CIDEr↑	BS↑
Segmentation threshold δ											
3°	70.2	88.7	**.596**	**.796**	.882	**2.24**	62.49	22.68	47.92	**55.20**	46.02
5° (default)	66.7	**90.1**	.584	.794	**.883**	2.35	63.45	22.67	47.80	53.16	45.58
10°	68.2	**90.1**	.544	.751	.846	2.56	62.47	22.40	47.25	50.76	44.04
15°	**70.5**	88.0	.580	.777	.862	2.38	**63.50**	**23.25**	**48.12**	53.74	**46.03**
Smoothing window w (frames)											
3	69.5	**90.8**	**.608**	.802	.886	**2.21**	63.03	22.66	47.78	54.03	45.84
5	**71.0**	89.1	.591	.791	.876	2.32	62.64	22.47	47.35	52.58	45.15
7 (default)	66.7	90.1	.584	.794	.883	2.35	**63.45**	**22.67**	47.80	53.16	45.58
11	66.7	**90.8**	**.608**	**.808**	**.890**	2.24	62.56	22.65	**48.04**	**55.53**	**46.05**
Trajectory position threshold τ_p (m)											
0.01	69.5	**91.7**	.527	.739	.833	2.54	62.76	22.40	47.70	50.69	44.46
0.03 (default)	66.7	90.1	**.584**	**.794**	**.883**	**2.35**	**63.45**	**22.67**	47.80	53.16	**45.58**
0.05	**70.2**	90.8	.543	.749	.831	2.55	62.34	22.27	47.39	50.19	44.08
0.10	68.7	90.8	.572	.781	.871	2.39	63.17	22.64	**48.19**	**54.04**	45.79

infer some aspects of global movement from joint angle patterns alone (e.g., cyclic hip and knee flexion implies walking), although explicit trajectory still provides a useful complementary signal.

Sensitivity to SMD conversion parameters. Table 4 tests the three SMD conversion parameters—angle segmentation threshold δ, smoothing window w, and trajectory position threshold τ_p. All three are stable: BABEL-QA varies by at most 4 points and captioning R@1 by at most 0.08 across tested settings. A few off-default values (e.g., $\delta = 3°$, $w = 11$) edge out the defaults on R@1/CIDEr, so light per-task tuning may yield modest gains. Overall, SMD is not a brittle hand-tuned prompt but a principled representation that degrades gracefully under parameter perturbation.

Zero-shot performance. Table 5 compares performance with and without LoRA fine-tuning. Without fine-tuning (zero-shot), the LLM achieves 35.6% on BABEL-QA and 31.7% on HuMMan-QA using Top-3 SMD, substantially above random chance (10% for 10 options). The answers are extracted with a lightweight regex that strips conversational prefixes before exact-string matching. This indicates that the LLM can partially interpret biomechanical descriptions in SMD without task-specific training. For captioning, the zero-shot model generates SMD-grounded but overly verbose descriptions, correctly identifying low-level movement components (e.g., lateral sway, torso rotation, arm swinging) but failing to summarize them into a concise action label—see Appendix B.2 for walking and waltz examples. This suggests that while the LLM can extract low-level motion cues from SMD without training, LoRA fine-tuning is needed to map biomechanical patterns to action-level semantics and to learn a concise captioning style.

4.4 Backbone Portability

Because SMD is plain text, switching the LLM backbone only requires retraining a lightweight LoRA adapter (∼40M parameters)—no motion encoder, projection layer are needed. To demonstrate

Table 5: Performance with and without LoRA fine-tuning on Qwen2.5-7B using Top-3 SMD.

Setting	BABEL-QA	HuMMan-QA	R@1↑	R@2↑	R@3↑	MMD↓	B@1↑	B@4↑	R-L↑	CIDEr↑	BS↑
Random guess	∼11.6	∼10.0	∼.031	∼.063	∼.094	—	—	—	—	—	—
Zero-shot	35.6	31.7	.075	.147	.221	6.86	19.76	1.14	18.45	0.86	8.26
Fine-tuned	**73.3**	**91.0**	**.547**	**.754**	**.842**	**2.59**	**61.74**	**22.25**	**47.52**	**52.28**	**44.94**

Table 6: Backbone portability across 8 LLMs from 6 model families (Top-3 SMD as input).

LLM	Params	BABEL-QA	HuMMan-QA	R@1↑	MMD↓	B@1↑	B@4↑	R-L↑	CIDEr↑	BS↑
Qwen2.5-3B	3B	64.6	89.3	.517	2.72	60.79	20.80	47.23	50.63	44.18
Qwen2.5-7B	7B	**73.3**	91.0	.547	2.59	61.74	22.25	47.52	52.28	44.94
Qwen2.5-14B	14B	71.5	88.7	.540	2.61	61.83	21.85	47.35	49.23	44.55
Qwen3-8B	8B	69.7	89.3	.541	2.49	60.75	**23.30**	**47.84**	**54.33**	**45.60**
Qwen3.5-9B	9B	72.8	**91.7**	.541	2.61	60.43	22.43	47.56	51.39	45.17
Gemma3-4B	4B	66.2	88.4	.552	2.55	**61.77**	22.85	47.64	52.26	45.28
LLaMA-3.1-8B	8B	63.1	89.1	.535	2.62	59.57	22.02	46.85	49.89	44.34
GLM-4-9B	9B	67.9	82.9	**.563**	**2.44**	60.86	22.75	47.58	53.20	45.10

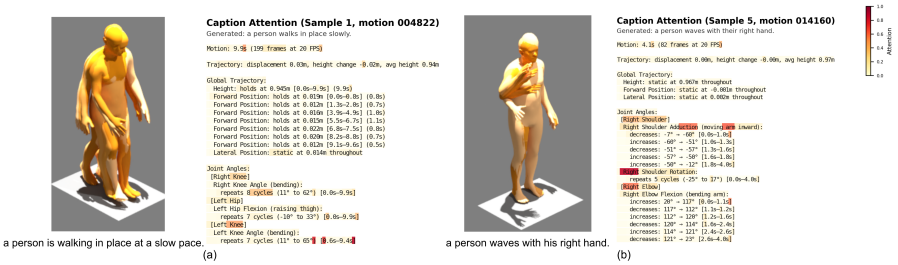


Figure 3: Attention heatmaps for two captioning examples. Background color on the SMD text indicates attention weight (yellow = low, red = high). (a) For "walking in place" the model attends to trajectory segments and cyclic joint angle patterns. (b) For "waving with right hand" the model focuses on the Right Shoulder and Right Elbow segments.

portability, we train 8 LLMs across 6 model families (3B–14B parameters) using the Top-3 SMD variant ($\sim$1,000 tokens/motion). Table 6 shows consistent performance across architectures: all models achieve $>$63% on BABEL-QA, $>$82% on HuMMan-QA, R@1 in 0.517–0.563, and CIDEr in 49.23–54.33. Within the Qwen2.5 family, scale helps (3B $\rightarrow$ 7B $\rightarrow$ 14B), and newer generations (Qwen3-8B, Qwen3.5-9B) match or surpass Qwen2.5-14B, consistent with more capable LLMs extracting more from the same SMD input; even the smallest model (Gemma3-4B) remains competitive. While switching backbone is convenient, the trade-off is a longer input sequence ($\sim$1,000–4,000 SMD tokens vs. $\sim$256 motion tokens for VAE/VQ-VAE methods), yielding roughly 1 sample/s inference on a single H200; see Appendix D.3 for the full cost comparison.

4.5 Interpretability

A distinctive advantage of SMD is its human-readable, text-based motion representation, which enables straightforward attention-based interpretability. We aggregate the attention weights over the SMD input across all layers of fine-tuned Qwen2.5-7B. Figure 3 shows heatmaps for two captioning examples. In both examples, the model selectively focuses on the SMD sections that are most relevant to the generated caption, demonstrating interpretable reasoning traces.

5 Conclusion

We have presented SMD, a rule-based conversion from human motion to structured text that enables LLMs to perform motion understanding without learned encoders or cross-modal alignment. With only lightweight LoRA fine-tuning, SMD surpasses prior state-of-the-art on both motion QA and captioning, transfers consistently across 8 LLMs from 6 families at 2–8 GPU-hours per adapter, and offers built-in interpretability via attention over human-readable tokens. The primary limitation is inference latency: the $\sim$4,000 tokens of All-26 SMD cause the inference time increased. The conversion also targets a fixed set of 26 angles over 22 SMPL joints and may miss finer articulation such as hand and finger motion. Promising directions include learned end-to-end SMD generation, motion-adaptive segmentation thresholds, and extending SMD to motion generation and editing.

AI USE STATEMENT

In this work, we havd used generative AI tools for coding and writing. They were used to literature survey; to implement methods; to give feedback on the design of experiments and ablations; and to assist with polish. We have not used generative AI tools to generate synthetic data sets. The remaining tasks with required disclosure are not applicable to this work.

Additionally, we also used generative AI tools to search the literature; suggesting the choice of baselines. We have reviewed all AI-assisted work. AI-written code was read by the authors before it was used for reported experiment. AI-drafted text was checked by the authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work studies human motion understanding from 3D skeleton sequences on public benchmarks. Because motion and gait patterns can be used to re-identify individuals, deployments should avoid monitoring specific groups without consent.

REPRODUCIBILITY STATEMENT

All conversion parameters, prompt templates, training hyperparameters, and dataset preparation steps are given in the Method section and in Appendices A.1, A.3, and D.1. Code is provided in the supplementary material.

REFERENCES

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*, pp. 6200–6211, 2025a.

Ling-Hao Chen, Shunlin Lu, Ailing Zeng, Hao Zhang, Benyou Wang, Ruimao Zhang, and Lei Zhang. Motionllm: Understanding human behaviors from human motions and videos. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025b.

Ginger Delmas, Philippe Weinzaepfel, Francesc Moreno-Noguer, and Grégory Rogez. Posefix: correcting 3d human poses with natural language. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 14972–14982. IEEE, 2023.

Ginger Delmas, Philippe Weinzaepfel, Thomas Lucas, Francesc Moreno-Noguer, and Grégory Rogez. Posescript: Linking 3d human poses and natural language. *IEEE transactions on pattern analysis and machine intelligence*, 47(7):5146–5159, 2024.

Mark Endo, Joy Hsu, Jiaman Li, and Jiajun Wu. Motion question answering via modular motion programs. In *International Conference on Machine Learning*, pp. 9312–9328. PMLR, 2023.

Qihang Fang, Chengcheng Tang, Bugra Tekin, Shugao Ma, and Yanchao Yang. Humocon: Concept discovery for human motion understanding. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7179–7190. IEEE, 2025.

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*, pp. 5152–5161, 2022a.

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*, pp. 580–597. Springer, 2022b.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

Biao Jiang, Xin Chen, Wen Liu, Jingyi Yu, Gang Yu, and Tao Chen. Motiongpt: Human motion as a foreign language. *Advances in Neural Information Processing Systems*, 36:20067–20079, 2023.

Junkun Jiang, Ho Yin Au, Jingyu Xiang, and Jie Chen. Lamogen: Language to motion generation through llm-guided symbolic inference. *arXiv preprint arXiv:2603.11605*, 2026.

Chen Li, Chinthani Sugandhika, Yeo Keat Ee, Eric Peh, Hao Zhang, Hong Yang, Deepu Rajan, and Basura Fernando. Imore: Implicit program-guided reasoning for human motion q&a. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 12987–12996, 2025a.

Kunhang Li, Jason Naradowsky, Yansong Feng, and Yusuke Miyao. How much do large language models know about human motion? a case study in 3d avatar control. *arXiv preprint arXiv:2505.21531*, 2025b.

Zhe Li, Weihao Yuan, Yisheng He, Lingteng Qiu, Shenhao Zhu, Xiaodong Gu, Weichao Shen, Yuan Dong, Zilong Dong, and Laurence Yang. Lamp: Language-motion pretraining for motion generation, retrieval, and captioning. In *International Conference on Learning Representations*, volume 2025, pp. 84238–84250, 2025c.

Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pp. 74–81, 2004.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916, 2023.

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*, pp. 851–866. 2023.

Thomas Markhorst, Zhi-Yi Lin, Jouh Yeong Chew, Jan van Gemert, and Xucong Zhang. Fine-grained human motion understanding with language models. *arXiv preprint arXiv:2606.20888*, 2026.

Jiawei Mo, Yixuan Chen, Rifen Lin, Yongkang Ni, Min Zeng, Xiping Hu, and Min Li. Mochat: Joints-grouped spatio-temporal grounding llm for multi-turn motion comprehension and description. *arXiv preprint arXiv:2410.11404*, 2024.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pp. 311–318, 2002.

Jacquelin Perry and Judith Burnfield. *Gait analysis: normal and pathological function*. CRC Press, 2024.

Kevin Schlegel, Lei Jiang, and Hao Ni. Using joint angles based on the international biomechanical standards for human action recognition and related tasks. *arXiv preprint arXiv:2406.17443*, 2024.

Amir Shahroudy, Jun Liu, Tian-Tsong Ng, and Gang Wang. Ntu rgb+ d: A large scale dataset for 3d human activity analysis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1010–1019, 2016.

Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4566–4575, 2015.

Yuan Wang, Di Huang, Yaqi Zhang, Wanli Ouyang, Jile Jiao, Xuetao Feng, Dan Xu, and Shixiang Tang. Motiongpt-2: A general-purpose motion-language model for motion generation and understanding. *IEEE Transactions on Circuits and Systems for Video Technology*, 2026.

Bizhu Wu, Jinheng Xie, Keming Shen, Zhe Kong, Jianfeng Ren, Ruibin Bai, Rong Qu, and Linlin Shen. Mg-motionllm: A unified framework for motion comprehension and generation across multiple granularities. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 27849–27858, 2025.

Ge Wu, Sorin Siegler, Paul Allard, Chris Kirtley, Alberto Leardini, Dieter Rosenbaum, Mike Whittle, Darryl D D'Lima, Luca Cristofolini, Hartmut Witte, et al. Isb recommendation on definitions of joint coordinate system of various joints for the reporting of human joint motion—part i: ankle, hip, and spine. *Journal of biomechanics*, 35(4):543–548, 2002.

Ge Wu, Frans CT Van der Helm, HEJ DirkJan Veeger, Mohsen Makhsous, Peter Van Roy, Carolyn Anglin, Jochem Nagels, Andrew R Karduna, Kevin McQuade, Xuguang Wang, et al. Isb recommendation on definitions of joint coordinate systems of various joints for the reporting of human joint motion—part ii: shoulder, elbow, wrist and hand. *Journal of biomechanics*, 38(5):981–992, 2005.

Yiming Wu, Wei Ji, Kecheng Zheng, Zicheng Wang, and Dong Xu. Mote: Learning motion-text diffusion model for multiple generation tasks. *arXiv preprint arXiv:2411.19786*, 2024.

Zhankai Ye, Bofan Li, Yukai Jin, Shuoqiu Li, Wei Wang, Yanfu Zhang, Shangqian Gao, and Xin Liu. Geomotiongpt: Geometry-aligned motion understanding with large language models. *arXiv preprint arXiv:2601.07632*, 2026.

Andy Zeng, Maria Attarian, Brian Ichter, Krzysztof Choromanski, Adrian Wong, Stefan Welker, Federico Tombari, Aveek Purohit, Michael Ryoo, Vikas Sindhwani, et al. Socratic models: Composing zero-shot multimodal reasoning with language. *arXiv preprint arXiv:2204.00598*, 2022.

Ce Zhang, Taixi Lu, Md Mohaiminul Islam, Ziyang Wang, Shoubin Yu, Mohit Bansal, and Gedas Bertasius. A simple llm framework for long-range video question-answering. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 21715–21737, 2024.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*, 2019.

Yao Zhang, Zhuchenyang Liu, Yanlan He, Thomas Ploetz, and Yu Xiao. Fine-grained motion retrieval via joint-angle motion images and token-patch late interaction. *arXiv preprint arXiv:2603.09930*, 2026.

Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 5745–5753, 2019.

Bingfan Zhu, Biao Jiang, Sunyi Wang, Shixiang Tang, Tao Chen, Linjie Luo, Youyi Zheng, and Xin Chen. Motiongpt3: Human motion as a second modality. *arXiv preprint arXiv:2506.24086*, 2025.

A Method Details

A.1 Complete list of joint angles

Our joint angle computation follows the biomechanical framework of Zhang et al. (2026); Schlegel et al. (2024); Wu et al. (2002; 2005), which defines angles by projecting bone vectors onto anatomical reference planes.

At each time frame, we establish a body-local coordinate frame using three reference points—the pelvis (origin), the left hip, and the right hip—following the convention $\mathbf{e}_x$ = forward, $\mathbf{e}_y$ = up, $\mathbf{e}_z$ = lateral (right-pointing): the lateral axis $\mathbf{e}_z$ is the unit vector from the left hip to the right hip; the vertical axis $\mathbf{e}_y$ is the unit vector from the hip midpoint to the pelvis, orthogonalized against $\mathbf{e}_z$; and the forward axis $\mathbf{e}_x = \mathbf{e}_y \times \mathbf{e}_z$ completes the right-handed system, with its sign chosen so that it points anteriorly. This body-local frame defines the subject's overall facing direction and serves as the root of the kinematic chain.

Individual joint angles are then computed within a hierarchy of joint-local coordinate frames, each defined relative to its parent joint along the kinematic chain. For example, hip angles are expressed in the pelvis frame; knee flexion in the hip (femur) frame; ankle angles in the knee (tibia) frame; shoulder angles in the lumbar-spine frame; and elbow flexion in the shoulder (upper-arm) frame. Expressing each angle in its parent frame follows standard ISB conventions (Wu et al., 2002; 2005) and isolates the motion of each joint from the orientation of upstream segments, so that, for instance, knee flexion measures only the bending of the knee regardless of how the hip or torso is oriented. The body-local root frame is derived from intrinsic body landmarks, and every joint-local frame is expressed relative to its parent along the kinematic chain. As a result, all joint angles encode only relative segment orientations and are invariant to the subject's global position and orientation.

The 26 angles are organized into 13 body part groups, listed below along with their parent reference frame (labeled in Figure 4):

1. **Pelvis** (3 angles, in the global frame): Tilt (forward/backward lean, projected onto the sagittal plane), List (lateral tilt, projected onto the coronal plane), and Rotation (yaw of the body forward direction).
2. **Lumbar Spine** (3 angles, in the pelvis frame): Extension, Lateral Bending, and Rotation, computed from the spine1-to-spine3 vector.
3. **Neck** (2 angles, in the lumbar-spine frame): Flexion (nodding) and Lateral Tilt, computed from the neck-to-head vector.

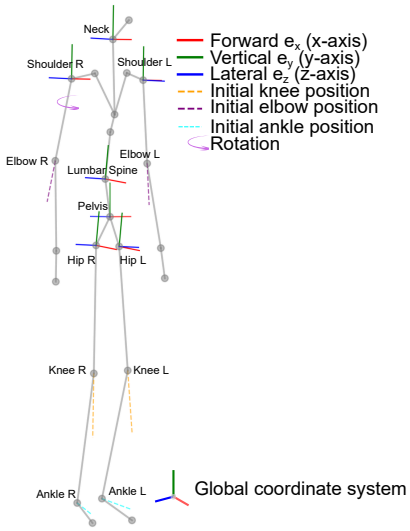


Figure 4: Illustration of the body-local and joint-local coordinate frames used for joint angle computation. The body-local frame (attached to the pelvis) is first established from the pelvis, left hip, and right hip, and defines the subject's overall facing direction. Joint-local frames are then attached to each of the 13 body part groups, organized into a kinematic chain so that each joint angle is expressed relative to its parent segment (e.g., knee flexion in the hip frame, elbow flexion in the shoulder frame). The three colored axes at each joint indicate forward (red), vertical (green), and lateral (blue). Dashed lines at the knees, elbows, and ankles show the initial (zero-flexion) positions against which flexion angles are measured, and the magenta arrow illustrates the rotation convention for joints with a twist degree of freedom (e.g., shoulder, hip). The global coordinate system (bottom-right) is shown for reference.

4. **Hip L/R** (3 angles each, in the pelvis frame): Flexion (sagittal plane), Adduction (coronal plane), and Rotation (femur twist).
5. **Knee L/R** (1 angle each, in the hip frame): Flexion, the angle between the femur and tibia vectors.
6. **Ankle L/R** (1 angle each, in the knee frame): Dorsi/plantarflexion, the angle between the tibia and foot vectors.
7. **Shoulder L/R** (3 angles each, in the lumbar-spine frame): Flexion (sagittal plane), Adduction (coronal plane), and Rotation (upper-arm twist).
8. **Elbow L/R** (1 angle each, in the shoulder frame): Flexion, the angle between the upper-arm and forearm vectors.

All angles are reported in degrees. Sign conventions follow standard biomechanical practice: flexion is positive, extension is negative; adduction is positive, abduction is negative. Wrappable angles (pelvis rotation, hip rotation, shoulder rotation) are unwrapped to avoid discontinuities at $\pm 180°$ before temporal segmentation. The pipeline assumes complete 3D joint positions from SMPL-format input; missing or noisy joints would need to be imputed upstream of SMD, and 2D inputs would require an off-the-shelf 3D pose lifter.

Axial twist via a secondary reference vector. A single 3D bone vector has only two directional degrees of freedom, so an axial twist around its longitudinal axis cannot be computed from the bone alone—a secondary reference vector pointing out of the bone axis is required. We construct a parent-segment-aligned local frame using the joint's flexion and adduction angles, then measure the angle of a secondary vector around the segment axis: hip rotation uses the tibia (knee-to-ankle) vector inside the femur-aligned frame; lumbar rotation uses the neck-to-right-shoulder vector inside the spine-aligned frame; and shoulder rotation uses the forearm (elbow-to-wrist) vector inside the upper-arm-aligned frame. In each case, the twist angle is the signed angle of the projection of the secondary vector onto the plane orthogonal to the segment's longitudinal axis.

Segmentation and cyclic detection. After calculated all 26 joint angles, each joint angle (and trajectory-axis) time series is first smoothed with a moving-average filter of window w, then segmented by peak/valley detection with a minimum amplitude δ, yielding an alternating sequence of "increases" and "decreases" intervals.

Table 7 lists the descriptive phrase that appears verbatim in the SMD text for each global trajectory axis and joint angle. The anatomical paraphrase in parentheses (e.g., "raising thigh" for hip flexion, "bending arm" for elbow flexion) makes each quantity immediately interpretable to an LLM relying on its pretrained world knowledge of human body mechanics, without any learned alignment.

Table 7: Descriptive phrases used in the SMD for each global trajectory axis and joint angle. The Global Trajectory block lists the verbs used for positive/negative/static segments along each axis. The Joint Angles block lists the display phrase that appears verbatim in the SMD text; each segment of a joint angle is additionally tagged with one of "increases", "decreases", "holds at", or "repeats N cycles" (e.g., "Left Hip Flexion (raising thigh): increases $3° \rightarrow 81°$ [0.0s–0.9s]"). The 13 body-part groups (shown as bracketed headers) match the section headers that appear in the assembled SMD string.

Global trajectory axis	**Direction verbs (positive / negative / static)**
Height	rises / lowers / holds at
Forward Position	moves forward / moves backward / holds at
Lateral Position	moves left / moves right / holds at
Body Rotation	turns left / turns right / holds at
Body-part group (13)	**Joint-angle display phrase (26)**
[Pelvis]	Pelvis Tilt (forward/backward lean)
	Pelvis List (lateral tilt)
	Pelvis Rotation (turning)
[Lumbar Spine]	Lumbar Extension (bending forward/backward)
	Lumbar Lateral Bending (side bend)
	Lumbar Rotation (twisting)
[Neck]	Neck Flexion (nodding)
	Neck Lateral Tilt (tilting head sideways)
[Right Hip]	Right Hip Flexion (raising thigh)
	Right Hip Adduction (moving leg inward)
	Right Hip Rotation
[Right Knee]	Right Knee Angle (bending)
[Right Ankle]	Right Ankle Angle (dorsi/plantarflexion)
[Left Hip]	Left Hip Flexion (raising thigh)
	Left Hip Adduction (moving leg inward)
	Left Hip Rotation
[Left Knee]	Left Knee Angle (bending)
[Left Ankle]	Left Ankle Angle (dorsi/plantarflexion)
[Right Shoulder]	Right Shoulder Flexion (raising arm forward)
	Right Shoulder Adduction (moving arm inward)
	Right Shoulder Rotation
[Right Elbow]	Right Elbow Flexion (bending arm)
[Left Shoulder]	Left Shoulder Flexion (raising arm forward)
	Left Shoulder Adduction (moving arm inward)
	Left Shoulder Rotation
[Left Elbow]	Left Elbow Flexion (bending arm)

A.2 Illustration of the SMD generation

As illustrated in Figure 5, we first compute the biomechanical joint angle of each joint from the raw motion, then segment each angle with our rule-based peak/valley segmentation, and finally convert the segments into SMD text according to the descriptive phrases in Table 7.

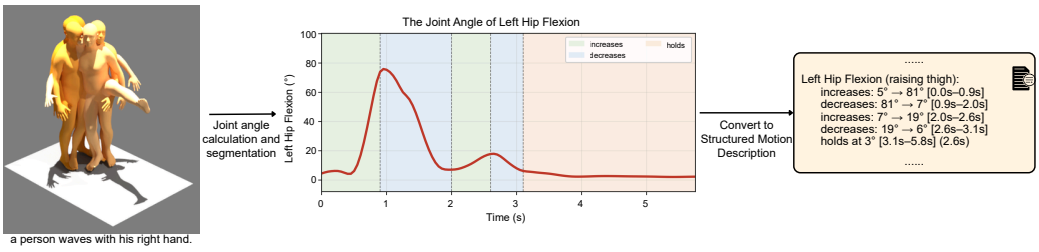


Figure 5: An example of the SMD generation procedure. Other joint angles and the global trajectory also follow the same procedure.

A complete All-26 SMD example for the motion "a person waves with his right hand" is shown in Figure 6. The full text contains 87 lines covering all 13 body part groups: most joints are static (gray), while the Right Shoulder and Right Elbow show active movement patterns (red/blue/orange) corresponding to the waving action. For the Top-K variant of SMD, joints are ranked by the peak-to-trough range $\max_t \theta_k^{(t)} - \min_t \theta_k^{(t)}$ over the motion, and the K joints with the largest ranges are kept.

A.3 Prompt structure

Figure 7 shows the full prompt template used during LoRA fine-tuning and inference. For motion QA (a), the prompt consists of a system instruction describing the task role, the SMD $\mathcal{S}$, the question text q, and the candidate answer options; the target output is the text of the correct option. For motion captioning (b), the prompt consists of a system instruction and the SMD $\mathcal{S}$; the target output is a natural-language caption describing the motion (e.g., "a person walks forward slowly"). During training, only the target response tokens contribute to the loss (all prompt tokens are masked).

B Additional Analyses

B.1 Open-vocabulary QA on BABEL-QA

The standardized 10-option format used in Table 1 bounds the answer to a discrete candidate set, which is convenient for comparing different methods. However, since MotionLLM (Chen et al., 2025b) does not provide checkpoints or training data, we were unable to retrain and evaluate MotionLLM (Chen et al., 2025b) using the 10-option versions of BABEL-QA and HuMMan-QA. Therefore, the comparison with MotionLLM in the table 1 is not fair. To enable a direct comparison with MotionLLM (Chen et al., 2025b), which reports open-vocabulary numbers in its original paper, we additionally evaluate our SMD-based approach on BABEL-QA and HuMMan-QA in a free-form generation setting: the prompt drops the candidate options entirely, and the model must generate the answer phrase autoregressively. We measure exact-match accuracy after a lightweight regex pass that strips conversational prefixes (e.g., "The answer is", "Option:"), lowercases the result, and removes leading/trailing punctuation, matching the protocol used for our zero-shot evaluation in Section 4. We use the same Qwen2.5-7B + LoRA checkpoint as in Table 6 without any re-fine-tuning and Top-3 joints SMD as input. This open-vocabulary evaluation is fair to all baselines: MotionLLM, IMoRe, and NSPose all train with the candidate options visible to the model, so removing the options at test time is a strictly harder regime that the comparison points were never optimized for.

Removing the candidate options at test time costs our model only 8.2 and 7.1 percentage points on BABEL-QA and HuMMan-QA respectively, and our open-vocabulary accuracy on BABEL-QA (65.1%) still exceeds the original MotionLLM number (43.6%) by more than 20 points. This indicates that LoRA-fine-tuned SMD does not rely on having the candidate options visible.

Full All-26 SMD Example: "a person waves with his right hand" (motion 014160)

```
Motion: 4.1s (82 frames at 20 FPS)

Trajectory: displacement 0.00m, height change -0.00m, avg height 0.97m

Global Trajectory:
  Height: static at 0.967m throughout
  Forward Position: static at -0.001m throughout
  Lateral Position: static at 0.002m throughout
  Body Rotation: holds at 6° [0.0s–4.1s] (4.1s)

Joint Angles:
 [Pelvis]
  Pelvis Tilt (forward/backward lean):
    static at -7° throughout
  Pelvis List (lateral tilt):
    static at 1° throughout
  Pelvis Rotation (turning):
    static at 6° throughout
 [Lumbar Spine]
  Lumbar Extension (bending forward/backward):
    static at 10° throughout
  Lumbar Lateral Bending (side bend):
    static at -3° throughout
  Lumbar Rotation (twisting):
    static at -5° throughout
 [Neck]
  Neck Flexion (nodding):
    holds at 46° [0.0s–4.1s] (4.1s)
  Neck Lateral Tilt (tilting head sideways):
    static at -1° throughout
 [Right Hip]
  Right Hip Flexion (raising thigh):
    static at 2° throughout
  Right Hip Adduction (moving leg inward):
    static at -9° throughout
  Right Hip Rotation:
    static at -12° throughout
 [Right Knee]
  Right Knee Angle (bending):
    static at 12° throughout
 [Right Ankle]
  Right Ankle Angle (dorsi/plantarflexion):
    static at -14° throughout
 [Left Hip]
  Left Hip Flexion (raising thigh):
    static at 3° throughout
  Left Hip Adduction (moving leg inward):
    static at -5° throughout
  Left Hip Rotation:
    static at -8° throughout
 [Left Knee]
  Left Knee Angle (bending):
    static at 8° throughout
 [Left Ankle]
  Left Ankle Angle (dorsi/plantarflexion):
    static at -20° throughout
 [Right Shoulder]
  Right Shoulder Flexion (raising arm forward):
    increases: -8° → 30° [0.0s–1.4s]
    holds at 29° [1.4s–2.9s] (1.4s)
    decreases: 28° → -12° [2.9s–4.0s]
  Right Shoulder Adduction (moving arm inward):
    decreases: -7° → -60° [0.0s–1.0s]
    increases: -60° → -51° [1.0s–1.3s]
    decreases: -51° → -57° [1.3s–1.6s]
    increases: -57° → -50° [1.6s–1.8s]
    increases: -50° → -12° [1.8s–4.0s]
  Right Shoulder Rotation:
    repeats 5 cycles (-25° to 17°) [0.0s–4.0s]
 [Right Elbow]
  Right Elbow Flexion (bending arm):
    increases: 20° → 117° [0.0s–1.1s]
    decreases: 117° → 112° [1.1s–1.2s]
    increases: 112° → 120° [1.2s–1.6s]
    decreases: 120° → 114° [1.6s–2.4s]
    increases: 114° → 121° [2.4s–2.6s]
    decreases: 121° → 23° [2.6s–4.0s]
 [Left Shoulder]
  Left Shoulder Flexion (raising arm forward):
    holds at -10° [0.0s–4.1s] (4.1s)
  Left Shoulder Adduction (moving arm inward):
    static at -7° throughout
  Left Shoulder Rotation:
    static at 8° throughout
 [Left Elbow]
  Left Elbow Flexion (bending arm):
    holds at 24° [0.0s–4.1s] (4.1s)
```

Figure 6: Complete All-26 SMD for motion 014160 ("a person waves with his right hand" 4.1s). Color coding: body part headers (dark red), increases (red), decreases (blue), static/holds (gray), repeats (orange), trajectory (green). The Right Shoulder and Right Elbow sections show active movement while all other joints remain static.

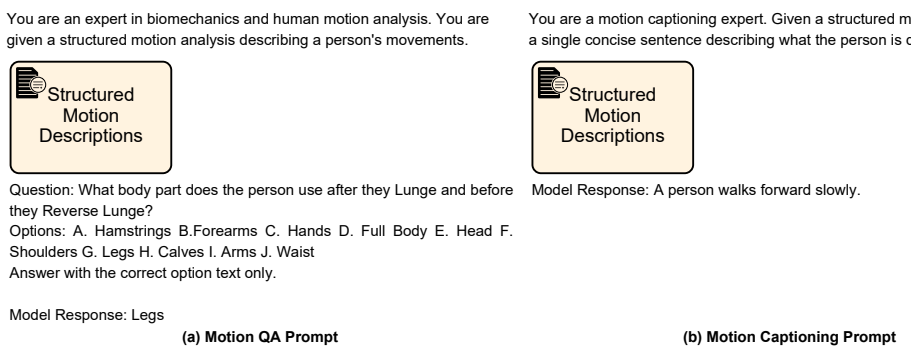


Figure 7: Prompt structure for (a) motion QA and (b) motion captioning.

Table 8: Open-vocabulary motion QA accuracy (%). †Number from the MotionLLM paper, which reports the original open-vocabulary protocol.

Method	Backbone	BABEL-QA	HuMMan-QA
MotionLLM† (Chen et al., 2025b) (open-vocab)	Vicuna-13B	43.6	—
Ours (10-option, for reference)	Qwen2.5-7B	73.3	91.0
Ours (open-vocab)	Qwen2.5-7B	65.1	83.9

B.2 Zero-shot captioning examples

Figure 8 illustrates the qualitative behavior of the zero-shot model (pre-fine-tuning) discussed in Section 4. For a walking motion (a), the zero-shot model correctly identifies the movement components (lateral sway, torso rotation, arm swinging) but produces a verbose description rather than the concise "a person walks in place." For a waltz (b), the model recognizes arm and leg movements but fails to identify the high-level action, describing it generically as "a complex dance or exercise routine" rather than a waltz. These examples show that the LLM can partially interpret SMD without training, but LoRA fine-tuning is needed to learn the concise captioning style and to map biomechanical patterns to action-level semantics.

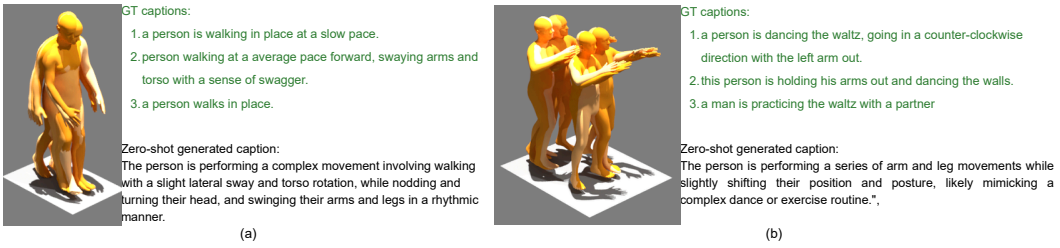


Figure 8: Zero-shot captioning examples. (a) Walking: the model identifies movement components (lateral sway, torso rotation, arm swinging) but produces a verbose description rather than the concise "a person walks in place." (b) Waltz: the model recognizes arm and leg movements but describes the action generically as "a complex dance or exercise routine" failing to identify the waltz.

B.3 Encoder domain adaptation for MotionGPT3-Qwen

The MotionGPT3-Qwen baseline uses a MotionGPT3 VAE that is frozen and pretrained on HumanML3D/AMASS. On HuMMan-QA, whose motions are reconstructed from a different (RGB-D) source, this frozen out-of-domain VAE performs poorly (22.0%). To test whether this failure reflects out-of-domain transfer rather than an inherent limitation of the encoder-based paradigm, we domain-adapt the VAE on HuMMan: we fine-tune its encoder and decoder by reconstruction on HuMMan

motions (reconstruction loss $0.26 \rightarrow 0.07$) and re-run the two-stage MotionGPT3-Qwen QA pipeline with the adapted VAE.

Table 9: Effect of domain-adapting the MotionGPT3 VAE on HuMMan-QA. Adapting the VAE on the target domain recovers most of the gap, but still trails SMD, which requires no per-domain retraining.

Method	HuMMan-QA
MotionGPT3-Qwen (frozen VAE)	22.0
MotionGPT3-Qwen (HuMMan-adapted VAE)	63.2
SMD (ours)	**90.1**

As shown in Table 9, adapting the VAE on HuMMan raises accuracy from 22.0% to 63.2%, recovering much of the gap. This indicates that the failure of the frozen baseline is largely a matter of out-of-domain transfer. Even so, the adapted encoder pipeline (63.2%) still trails SMD (90.1%), and, crucially, SMD reaches 90.1% with no per-domain encoder retraining at all: its rule-based conversion is domain-invariant by construction, which is the practical advantage of the encoder-free design.

B.4 BENCHMARK-PRIOR CONTROLS

To quantify how much of the QA accuracy reflects fine-grained motion reasoning rather than dataset priors, we run three controls. (i) Text-only: the same Qwen2.5-7B + LoRA trained and evaluated on the question and options only, with no motion. (ii) Wrong-motion: at test time each question is paired with another motion's SMD without retraining the model. (iii) Shuffled: we shuffle the SMD by lines without retraining the model. The global trajectory may appear after the joints, and the order of the joints may change, but the content of each joint angle remains unchanged.

Table 10: No-motion and benchmark-prior controls.

Input	BABEL-QA	HuMMan-QA
Text-only (no motion)	20.4	18.6
Wrong-motion SMD	33.8	27.2
Shuffled SMD	54.7	72.6
SMD (ours)	**66.7**	**90.1**

The Text-only model scores $20.4/18.6\%$, which is above random (about 11.6 / 10%), so some benchmark prior does exist. But these scores are far from high on either dataset, so the benchmarks are not substantially solvable only from question and option priors. SMD lifts accuracy to 66.7 / 90.1%, a gain of +45 / +72 points that can only come from reading the motion.

Feeding a wrong motion SMD makes accuracy go back to the "Text-only" level (HuMMan 90.1 $\rightarrow$ 27.2, and BABEL 66.7 $\rightarrow$ 33.8). The text-only and wrong motion results demonstrate that the performance improvement achieved by our methods is primarily from motion reasoning. Shuffling lines, which keeps which joints move but destroys order, degrades accuracy mildly (HuMMan 90.1 $\rightarrow$ 72.6).

B.5 ROBUSTNESS TO INPUT PERTURBATIONS

Because there is no paired dataset of noisy real-world skeletons with corresponding text captions and QA pairs, to validate the stability of our method under noisy data, we conduct perturbation experiments: we perturb the raw 3D joints before SMD conversion and evaluate our checkpoint without retraining. We test Gaussian joint jitter at $\sigma \in \{1, 2, 5\}$ cm, joint occlusion (2 or 4 joints), and frame-rate change (15 or 30 fps).

Occlusion and frame-rate change are nearly harmless (at most 2.5 points on HuMMan). Jitter of 1 to 2 cm costs only 4 to 9 points, and 5 cm jitter leaves HuMMan at 73.9%, still far above the 10% random guess level.

Table 11: Robustness of the Top-3 SMD model to input perturbations (QA accuracy, %).

Perturbation	BABEL-QA	HuMMan-QA
Clean	73.3	91.0
Jitter $\sigma = 1$ cm	68.7	86.7
Jitter $\sigma = 2$ cm	64.9	81.6
Jitter $\sigma = 5$ cm	53.9	73.9
Occlusion, 2 joints	71.8	90.2
Occlusion, 4 joints	69.7	88.6
Frame-rate 15 fps	71.8	91.0
Frame-rate 30 fps	71.3	90.6

B.6 Token compression study

All-26 SMD is long (∼4,000 tokens). We study three lightweight strategies to control input length. Top-K selection keeps only the K joints with the largest angular displacement. SMD-Compact is a rule-based compressor that folds static joints into a single summary line and compacts redundant segment text, without dropping any active joint. LLM-guided adaptive selection is a leakage-free scheme in which the model, from the question and options alone, selects which body-part channels to read, and then answers from only those channels; it is therefore specific to QA.

Table 12: Token compression for motion QA (accuracy, %).

Strategy	Input tokens	BABEL-QA / HuMMan-QA
All-26 (default)	∼4,000	66.7 / 90.1
Top-3 selection	∼1,000	73.3 / 91.0
SMD-Compact	-14 to -34%	64.6 / 88.6
LLM-guided adaptive	-49.5%	60.5 / 81.7

Top-3 removes about 75% of the input tokens while improving QA accuracy over All-26 ($73.3/91.0$ versus $66.7/90.1$). SMD-Compact and LLM-guided adaptive selection reduce tokens further with only a small accuracy cost. The same compression also applies to motion captioning (Table 13), where the LLM-guided scheme does not apply because there is no question.

Table 13: Token compression for motion captioning on the HumanML3D test set.

Method	R@1↑	R@2↑	R@3↑	MMD↓	B@1↑	B@4↑	R-L↑	CIDEr↑	BS↑
Top-3	.547	.754	.842	2.59	61.74	22.25	47.52	52.28	44.94
SMD-Compact	.573	.791	.869	2.40	61.63	22.39	47.55	51.30	43.97
Ours (All-26)	.584	.794	.883	2.35	63.45	22.67	47.80	53.16	45.58

Both Top-3 and SMD-Compact stay close to the full All-26 model across all captioning metrics while using fewer tokens.

C Other Use Cases

C.1 Cross-skeleton generalization (NTU RGB+D 60)

The 22-joint SMPL skeleton used in Section 3 is just one of several common motion representations. To probe whether SMD generalizes beyond this topology, we apply it to NTU RGB+D 60 (Shahroudy et al., 2016), a Kinect-v2 dataset with 25 joints per frame and 60 action classes. NTU is a strong test of generalization for two reasons: it uses a different sensor (Kinect depth, not optical MoCap) and a different skeleton topology (25 vs. 22 joints, with NTU including hand-finger joints absent in our SMPL subset). We use the X-Sub split, which separates training and test subjects.

NTU's 25-joint skeleton already shares a hierarchical structure with SMPL: 19 of the 25 NTU joints have a direct SMPL counterpart (knees, ankles, shoulders, elbows, wrists, hips, neck, head, etc.). Three additional SMPL joints absent from NTU—spine1, left/right collar—are interpolated as midpoints of nearby NTU joints (SpineBase–SpineMid for spine1, Neck–Shoulder for the collars). The four NTU finger and thumb joints are dropped. We additionally flip the lateral axis sign to convert NTU's right-handed camera frame (x pointing to the subject's right) into the HumanML3D convention used by SMD, and downsample from 30 FPS to 20 FPS to match SMD's training-time frame rate.

A subtle but important detail concerns the SMPL pelvis joint. In SMPL/HumanML3D the pelvis sits roughly 9 cm above the hip midpoint along the spine, so the up-direction $\mathbf{e}_y \propto \text{pelvis} - \text{hip_mid}$ used in f_{SMD}'s body-local frame (Section 3.1, Appendix A.1) is a clean, well-conditioned vector. NTU's SpineBase joint, by contrast, sits essentially at the hip midpoint (≈ 1 mm offset on a typical motion), so mapping SpineBase directly to the SMPL pelvis would make this difference vector dominated by Kinect noise and corrupt every downstream joint angle. We therefore map the SMPL pelvis to NTU's SpineMid (joint 1, ≈ 26 cm above the hips), which restores the same well-conditioned up direction and produces angle distributions visually consistent with HumanML3D-derived motion. No other modification to f_{SMD} is required.

We frame action recognition as a 10-option QA: "What action is the person performing? Options: A. . . . J."—9 distractor classes are sampled from the remaining 59 NTU classes per query (fixed seed) plus the correct one. We compare two regimes: (i) **Zero-shot**: we apply our All-26 SMD checkpoint trained on BABEL-QA + HuMMan-QA (Table 1), with no NTU-specific training. (ii) **Fine-tuned (1 epoch)**: we generate SMD All-26 for the 38,165 training motions of NTU60 X-Sub and continue LoRA fine-tuning the same Qwen2.5-7B base for one epoch, using the same hyperparameters as Section 4.

Table 14: Action recognition on NTU RGB+D 60 (X-Sub split, 10-option format). Random-guess accuracy is 10%. The zero-shot row uses our BABEL/HuMMan-trained checkpoint with no NTU exposure; the fine-tuned row uses the same backbone with one epoch of LoRA fine-tuning on the NTU train split.

Setting	Training data	Top-1 (%)
Random guess	—	10.0
Ours (SMD All-26, zero-shot)	BABEL-QA + HuMMan-QA	20.5
Ours (SMD All-26, 1-epoch FT)	NTU60 training set	**83.8**

Discussion. First, the zero-shot number (20.5%, $\sim 2\times$ chance) is modest compared with our in-distribution numbers in Table 1. A per-class breakdown explains why: the model performs strongly on broad locomotor and gestural actions that resemble its BABEL/HuMMan training distribution—e.g., *throw* and *hopping* reach 100%, *jump up*, *kicking something*, *staggering*, *falling*, and *walking apart from each other* reach 80%—but collapses to 0% on two failure groups: (a) everyday activities never seen in pre-training, such as *eat meal*, *brushing teeth*, *wear jacket*, *put on glasses*, *check time from watch*, *typing on keyboard*, and *reading*; and (b) activities that are intrinsically hard to disambiguate from skeleton alone because they hinge on object interaction or fine-grained finger movement, such as *tear up paper*, *wipe face*, *touch head/neck/back*, and *pat on back of other person*. The first group reflects a real limitation of our pre-training corpus: training only on BABEL-QA and HuMMan-QA exposes the LLM to a relatively narrow action vocabulary (locomotion + body-part gestures + fitness), and large-scale multi-task fine-tuning across more diverse activity datasets is a natural avenue for future work.

Second, after just one epoch of LoRA fine-tuning on the NTU60 X-Sub training split, the same model reaches 83.8% top-1 on the held-out test split, a 63-point jump over zero-shot. The SMD conversion pipeline, prompt template, and architecture are all unchanged from Section 4; only the LoRA adapter weights are updated. This confirms two things: (i) the SMD biomechanical text representation transfers to a different skeleton format and acquisition pipeline once the LLM is exposed to the new action vocabulary, and (ii) SMD's conversion stack is data-efficient, since a single epoch on a non-AMASS, non-MoCap dataset is enough to recover most of the gap on previously unseen everyday actions.

C.2 FINE-GRAINED MOTION CAPTIONING

MG-MotionLLM (Wu et al., 2025) introduces a fine-grained motion-to-detailed-text (m2dt) task in which the caption must describe per-body-part movements over time rather than only the overall action label. SMD is a natural fit for this task: its per-joint, per-segment text already carries the fine-grained structure that the task requires, so the LLM can directly transcribe and combine joint-level information rather than having to recover it from a compressed motion latent.

We evaluate on the FineMotion test set (Wu et al., 2025), which re-annotates each HumanML3D motion with detailed body-movement descriptions over time. Each motion sequence is divided into fixed-length snippets of 0.5 s, and each snippet is paired with an automatically generated, fine-grained description of the movements of individual body parts during that interval (e.g., "*0.0s–0.5s: Move your right hand to your right shoulder. 0.5s–1.0s: Move your right elbow down to your chest. Extend your left arm out to the side slightly.*"). Snippets are joined into a single ground-truth document for the whole motion via a special ⟨SEP⟩ token; empty snippets are replaced by ⟨Motionless⟩. In total, FineMotion contains 420,968 snippets across the HumanML3D motion set.

Following MG-MotionLLM (Wu et al., 2025), we report m2dt accuracy at two levels: (i) **Sequence level**: the model emits the full multi-snippet detailed text for the whole motion as a single document, and linguistic metrics (BLEU, ROUGE, BERTScore) are computed against the ground-truth document. (ii) **Snippet level**: the predicted and ground-truth documents are each split by ⟨SEP⟩ into per-snippet descriptions, and the metrics are computed snippet-by-snippet and then averaged. The snippet-level score is harder than the sequence level because errors in any single 0.5 s window are penalized individually rather than smoothed over the full motion.

The detailed-text format used by FineMotion is structurally different from the free-form HumanML3D captions our model was trained on in Section 4, so we train a dedicated LoRA adapter for the m2dt task on the FineMotion training split, keeping the SMD conversion (All-26, Qwen2.5-7B base, identical hyperparameters to Section 4) and the LoRA configuration unchanged. The SMD pipeline itself is rule-based and is not re-trained. We compare against MG-MotionLLM at its largest reported size (T5-Large, the strongest configuration in Table 4 of (Wu et al., 2025)), which is also trained on FineMotion (within their 28-task multi-granularity scheme), giving a fair head-to-head on the same data.

Table 15: Fine-grained motion captioning on the FineMotion test set, evaluated under the m2dt protocol of MG-MotionLLM (Wu et al., 2025) at both the sequence and snippet levels. Higher is better for all metrics. MG-MotionLLM numbers are taken from their Table 4 (T5-Large variant).

Level	Method	B@1↑	B@4↑	B@7↑	ROUGE↑	BERTScore↑
Sequence	MG-MotionLLM (T5-Large) (Wu et al., 2025)	81.73	66.29	53.56	65.8	52.4
	Ours (SMD All-26)	86.95	69.01	55.28	66.1	56.1
Snippet	MG-MotionLLM (T5-Large) (Wu et al., 2025)	68.25	49.26	37.07	61.4	52.2
	Ours (SMD All-26)	69.94	51.33	40.07	62.6	56.9

SMD outperforms MG-MotionLLM on every metric at both evaluation levels. At the sequence level, BLEU@1 improves by $+5.22$ points ($81.73 \rightarrow 86.95$) and BERTScore by $+3.7$, indicating more accurate transcription of the per-body-part movement vocabulary. The gap is preserved at the harder snippet level—where errors in any single 0.5 s window are penalized individually—with BERTScore actually growing to $+4.7$, suggesting that SMD's predictions remain semantically faithful even at fine temporal granularity. We attribute this to SMD's structural fit with the FineMotion format: SMD's per-joint, time-localized segments already provide exactly the kind of body-part-and-time-indexed information the m2dt task requires, so the LLM only has to learn to phrase it in MG-MotionLLM's snippet template, rather than recover the joint-level structure from a compressed motion latent.

D EXPERIMENTAL SETUP AND COSTS

D.1 DATASETS, BASELINES, AND 10-OPTION STANDARDIZATION

Original benchmarks. BABEL-QA (Endo et al., 2023) defines three question types over BABEL/AMASS motions: action queries ("What action does the person do after X?"), body part queries

("What body part does the person use?"), and direction queries ("What direction does the person move?"). The original benchmark uses all valid labels as options, resulting in 4 options for direction, 8 for body part, and 20 for action. HuMMan-QA (Li et al., 2025a) follows the same format but uses the HuMMan dataset, which has a much larger label vocabulary: up to 155 action labels, 18 body parts, and 6 directions.

10-option standardization. The varying number of options (4–155) makes cross-method comparison difficult, as random-guess accuracy ranges from 25% (4 options) to 0.6% (155 options). We standardize all questions to at most 10 options: for questions with more than 10 original options, we randomly sample 9 distractors plus the correct answer; for questions with fewer than 10 options, we retain the original set. This yields an average of 8.6 options for BABEL-QA and 10.0 for HuMMan-QA, with random-guess accuracy of approximately 11.6% and 10.0% respectively. The 10-option question files are generated once and shared across all methods for fair comparison.

Zero-shot QA answer extraction. The zero-shot QA results in Table 5 use the same evaluation protocol as the fine-tuned runs (exact-string matching against the candidate option text) plus a lightweight regex pass that strips conversational prefixes (e.g., "The answer is", "Option:", "A.", leading/trailing punctuation) and lowercases the prediction before comparison. This pass is applied uniformly to every method's outputs. The regex normalization is necessary because base LLMs without task-specific fine-tuning often wrap the answer in a sentence (e.g., "The correct option is left hand."); without it, exact-string matching would penalize formatting habits rather than reasoning ability. The same regex is reused in the open-vocabulary evaluation of Appendix B.1.

Baseline retraining. All QA baselines reported in Table 1 expect MotionLLM (Chen et al., 2025b) are retrained and evaluated on the standardized 10-option format:

IMoRe (Li et al., 2025a) is a specialized motion QA model that uses classifier heads over a fixed label vocabulary. We retrain IMoRe on the 10-option training data for both BABEL-QA and HuMMan-QA. At evaluation, we extract the full logit vector from the classifier, mask it to only the 10 candidate options for each question, and take the argmax over the masked logits. This ensures the model selects from exactly the same option set as all other methods.

MotionLLM (Chen et al., 2025b) results are taken from the original paper, as neither the training data nor the checkpoints are publicly available for retraining. The reported 43.6% on BABEL-QA uses the original option format and is therefore not directly comparable to our 10-option results, though we include it as a reference. MotionLLM uses Vicuna-13B as backbone with a learned motion encoder that projects SMPL motion features into the LLM's embedding space via a linear projection layer, trained with LoRA on a combined video-motion dataset (MoVid).

NSPose (Endo et al., 2023) is the original motion QA method that defines the task and proposes a neuro-symbolic framework. It uses a pretrained Motion ViT encoder and executes modular programs recursively over the learned motion representations. We retrain NSPose on the 10-option format using their released code.

Captioning baselines. For the captioning experiments in Table 1, baseline numbers for TM2T, MotionGPT, LaMP, MoTe, and MotionGPT3 are taken from the MotionGPT3 paper (Zhu et al., 2025), which evaluates all methods under the same protocol on HumanML3D, except for CIDEr which we re-compute on the released checkpoints with the cleaned `pycocoevalcap` pipeline described in Appendix D.2 (the original numbers were inflated by extraneous symbols such as colons and newlines in the predictions). MG-MotionLLM (Wu et al., 2025) numbers are from their CVPR 2025 paper.

MotionGPT3-Qwen. To provide a controlled comparison isolating the effect of motion representation, we replicate the MotionGPT3 paradigm on the same Qwen2.5-7B backbone used by our method. Specifically, we use the pretrained MotionGPT3 VAE encoder (frozen) to encode each motion into a continuous latent vector of dimension 256. This latent is projected to N LLM-sized token embeddings via a two-layer MLP ($256 \to 2048 \to N \times 3584$, where 3584 is the Qwen2.5-7B hidden dimension). Training follows the two-stage LLaVA (Liu et al., 2023) protocol: in stage 1, the LLM is frozen and only the projection MLP is trained; in stage 2, LoRA is added to the LLM and both the MLP and LoRA are trained jointly. Both stages use learning rate 1e-4 (stage 1) and 1e-4 (stage 2), with 5 epochs each. For QA, the motion tokens are prepended to the question in the prompt;

for captioning, they replace the SMD text. We evaluate configurations with $N \in \{4, 32, 64, 128\}$ projection tokens (see Appendix D.4).

D.2 Evaluation metric definitions

We evaluate motion-to-text captioning using two categories of metrics.

Text-motion alignment metrics. These metrics assess whether the generated caption semantically matches the corresponding motion, using the pretrained T2M evaluator (Guo et al., 2022a) which provides a shared embedding space for text and motion.

R-Precision (R@k, $k \in \{1, 2, 3\}$) measures retrieval accuracy: given a pool of 32 motions, the generated caption is used to retrieve the best-matching motion via cosine similarity in the T2M embedding space. R@k reports the fraction of times the ground-truth motion appears in the top-k retrieved results, averaged over all test samples. Higher R-Precision indicates better semantic alignment between the generated text and the original motion.

MM-Distance (Multi-Modal Distance) computes the average Euclidean distance between the T2M embeddings of each generated caption and its ground-truth motion. Lower MM-Distance indicates that generated captions are semantically closer to their corresponding motions in the learned embedding space.

Both metrics are computed using the Comp_v6_KLD01 checkpoint of the T2M evaluator.

Linguistic metrics. These metrics evaluate the textual quality of generated captions by comparing them against reference captions using standard NLP evaluation tools.

BLEU@k (Papineni et al., 2002) measures modified n-gram precision between the generated and reference captions, with k indicating the maximum n-gram order. We report BLEU@1 (unigram precision, capturing word overlap) and BLEU@4 (up to 4-gram precision, capturing phrase-level similarity).

ROUGE-L (Lin, 2004) computes the longest common subsequence (LCS) between the generated and reference captions, measuring recall-oriented overlap at the sentence level.

CIDEr (Vedantam et al., 2015) uses TF-IDF weighted n-gram matching to evaluate consensus between the generated caption and multiple reference captions, designed specifically for image/video captioning evaluation.

BERTScore (Zhang et al., 2019) computes token-level cosine similarity between contextual embeddings of the generated and reference captions using a pretrained BERT model, capturing semantic similarity beyond surface-level n-gram overlap.

All linguistic metrics are computed on motions with 20–200 frames using pycocoevalcap, matching the data range used by MotionGPT3 (Zhu et al., 2025).

CIDEr implementation. Since the original CIDEr computation in prior works inadvertently includes extraneous symbols such as colons and newlines, which affect the scores. We compute CIDEr using the `pycocoevalcap` library (the standard COCO Caption evaluation toolkit), which implements CIDEr-D from Vedantam et al. (Vedantam et al., 2015). The implementation uses n-grams up to n=4 with a Gaussian length-penalty standard deviation of σ=6.0, and computes IDF statistics directly from the reference captions of the evaluation subset. Following standard practice, when a motion has multiple reference captions we treat them as a set of references for the same query, rather than averaging individual scores. We note that CIDEr is highly sensitive to corpus-level IDF statistics and to caption tokenization; minor differences in the evaluation subset (e.g., length-filtered vs. full test set) or in the underlying tokenizer can shift the absolute CIDEr value by several points without affecting the relative ordering of methods. For this reason, we report all baseline and SMD CIDEr numbers computed with the same `pycocoevalcap` pipeline on the identical evaluation subset, ensuring a fair comparison even if the absolute values differ from those reported in the original papers.

D.3 TRAINING AND INFERENCE COST COMPARISON

This appendix details the training and inference costs of SMD relative to encoder-based motion LLMs summarized in Section 4.4. Table 16 lists the head-to-head comparison.

Training cost. Retraining SMD on a new LLM backbone takes 2–8 GPU-hours for QA and 6–20 hours for captioning on a single H200, with the captioning range varying by backbone size (3B–14B parameters) and SMD variant (Top-3 ≈1k tokens vs All-26 ≈4k tokens); the ∼20 GPU-hour figure quoted in Section 4 corresponds to the All-26 + Qwen2.5-7B reference run reported in Table 1. Only ∼40M LoRA parameters are updated while the base LLM is frozen and the SMD conversion itself is rule-based and has no trainable parameters. In contrast, encoder-based methods such as MotionGPT3 and MG-MotionLLM retrain the full pipeline on each new LLM, including the motion encoder and the alignment module. This involves hundreds of millions of parameters and a multi-stage schedule (pretrain the motion encoder, then the projection or alignment module, then the LLM), on the order of tens of GPU-hours even for their relatively small backbones.

Inference cost. The main trade-off of SMD is a longer input sequence. Encoder-based methods compress a motion into ∼256 tokens, whereas SMD produces ∼1,000 tokens for the Top-3 variant and ∼4,000 tokens for the All-26 variant. We reported the cost comparision in Table 16. On a single H200 GPU, the average latency of our approach for captioning task is 915 ms/sample (1.1 samples/s) for Top-3 SMD and 1,154 ms/sample (0.9 samples/s) for All-26 SMD, using approximately 15.5 GB of GPU memory. To isolate the input-length effect from the backbone size, we also benchmark the same-backbone MotionGPT3-Qwen baseline (Qwen2.5-7B + LoRA + frozen MotionGPT3 VAE + MLP projection, $N{=}4$ motion tokens) under the identical protocol; it averages 300 ms/sample (3.3 samples/s) at 14.7 GB GPU memory.

Table 16: Comparison of training and inference costs across LLM backbones. Our method retrains only a lightweight LoRA adapter, whereas prior methods must retrain the full alignment pipeline.

	Ours (SMD)	MotionGPT3	MG-MotionLLM
LLM backbone	Any (8 tested)	GPT-2 (124M)	T5-Base (220M)
Motion representation	Rule-based text	Learned VAE	Learned VQ-VAE
Training stages	1 (LoRA only)	3	2
Retrain on new LLM	LoRA only	Full pipeline	Full pipeline
Trainable params	∼40M	∼124M	∼220M
Caption train time	∼6–20 h	∼60 h*	∼60 h*
Input tokens (avg)	∼1,000–4,000	∼256	∼256
Inference (ms/sample)	915–1,154	<100 (smaller LLM)	

*Estimated total multi-stage training time including encoder pretraining.

D.4 MOTIONGPT3-QWEN: PROJECTION TOKEN ABLATION

Table 17 reports the full token number ablation for MotionGPT3-Qwen on both QA and captioning. As the number of projection tokens increases from 4 to 128, the projection MLP grows from 15M to 940M parameters. For QA, all configurations achieve similar BABEL-QA accuracy (47–50%), far below SMD (66.7%), and perform poorly on HuMMan-QA (15–24% vs. SMD 90.1%) due to the cross-domain issue discussed in Section 4 and Appendix B.3. For captioning, performance holds steady at 4 and 32 tokens but degrades with larger MLPs (64 and 128 tokens), as the projection layer maybe overfits on the limited training data.

E QUALITATIVE ANALYSIS

This section provides side-by-side qualitative comparisons of our SMD-based approach against the baselines on motion QA and motion captioning.

Table 17: MotionGPT3-Qwen with varying projection tokens. All configurations use the same frozen VAE and two-stage training.

Tokens	Params	BABEL-QA	HuMMan-QA	R@1↑	R@2↑	R@3↑	MMD↓	B@1↑	B@4↑	R-L↑	CIDEr↑	BS↑
4	30M	50.1	22.0	.555	.753	.843	2.47	59.43	19.60	45.82	46.13	42.87
32	235M	50.4	23.5	.570	.759	.846	2.44	57.22	17.53	44.31	43.52	41.76
64	470M	49.9	14.6	.547	.739	.826	2.60	56.09	17.25	42.38	37.89	37.86
128	940M	47.3	23.6	.500	.710	.803	2.65	59.33	19.78	44.73	41.68	40.74
SMD (ours)		**66.7**	**90.1**	**.584**	**.794**	**.883**	**2.35**	**63.45**	**22.67**	**47.80**	**53.16**	**45.58**

E.1 MOTION QA

Figure 9 shows four motion QA examples spanning the three query types (direction, body part, action) and both datasets (BABEL-QA, HuMMan-QA). On all four cases SMD picks the correct answer while both IMoRe and MotionGPT3-Qwen pick a wrong but semantically related option—e.g., *forward* or *right* instead of *left*, *left arm/hand* instead of *right arm*, *catch* or *throw* instead of *jumping jacks*, and *Half Squat* or *place something* instead of *Arm Curl*. The grounded, body-part-indexed text that SMD provides makes it easy for the LLM to attend to the discriminative joint.

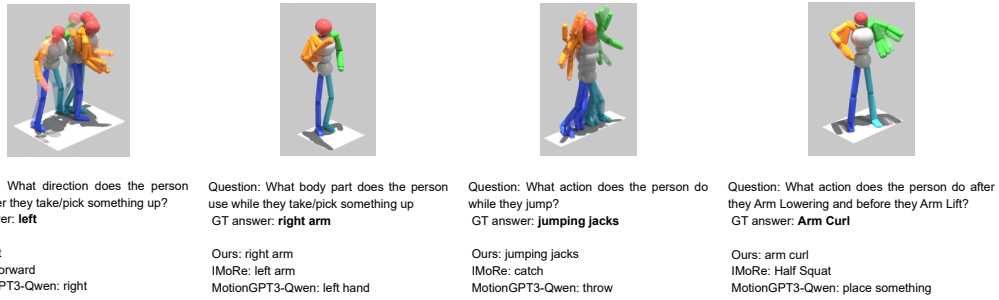


Figure 9: Qualitative motion QA comparison on BABEL-QA and HuMMan-QA. Each panel shows a representative frame of the motion, the question, the ground-truth answer, and the predictions from IMoRe (Li et al., 2025a), MotionGPT3-Qwen, and our SMD-based approach. SMD answers all four examples correctly; both baselines confuse the discriminative body part, direction, or action category in every case.

E.2 MOTION CAPTIONING

Figure 10 shows four motion captioning examples from the HumanML3D test set comparing our SMD model against MotionGPT3 (Zhu et al., 2025) and MG-MotionLLM (Wu et al., 2025). On all four examples SMD produces detailed, body-part-grounded captions that reflect the multi-step structure of the references—e.g., *"raises their right hand above their head and then their left hand, while rocking back and forth"* for the pantomime gesture, *"steps to their right, then back to the left, and finally back to the right"* for the side-step sequence, and *"throws something with their right hand and then catches something with both hands"* for the throwing motion. The two baselines, lacking this body-part trace, often fall back on generic descriptions (*"standing still"*, *"slightly moved forward"*, *"walking around"*), replace the action with a visually similar one (*"kicking something"*, *"kicks the left leg"*), or recover only the coarse action label without its discriminative body part or temporal structure (*"throws an object"*).



references:
"a person lifts his left hand up, then right hand up and waves them about, then put his hands down",
"a person is being a mime",
"a person makes pantomime hand gestures above his head with both hands."

Ours: a person raises their right hand above their head and then their left hand, while rocking back and forth on their feet.
MotionGPT3: a person is standing still.
MG-MotionLLM: person looks like they are doing nothing but lifting their legs.

references:
"a person side steps to the right and then to the left"
"a person side steps to the right, then side steps to the left and final back to the right",
"a person walking to the right front , then stepping back to center, walking back to the right front again"

Ours: a person steps to their right, then back to the left, and finally back to the right.
MotionGPT3: a person slightly moved forward.
MG-MotionLLM: a person is kicking something.

references:
"a person placing something in their hand and then throwing it while twisting their body.",
"the person is throwing a ball with his right hand.",
"a person throws an object with his right hand, then holds both hands up over his head"

Ours: a person throws something with their right hand and then catches something with both hands.
MotionGPT3: a person is walking around.
MG-MotionLLM: a person throws an object

references:
"a person jumps forward with the right foot, moves the left foot in front of the right foot, and then moves both feet back to standing position.",
"a person hit something with right hand",
"a person is standing, then quickly take a step forward and to the right, while making a small kicking motion with their left leg."

Ours: a person steps forward with their right foot and hits something with their right hand.
MotionGPT3: a person kicks the left leg
MG-MotionLLM: a person is standing still

Figure 10: Qualitative motion captioning comparison on HumanML3D. Each panel shows a representative motion frame, the ground-truth reference captions, and the captions produced by MotionGPT3 (Zhu et al., 2025), MG-MotionLLM (Wu et al., 2025), and our SMD-based approach. SMD captures the multi-step, body-part-specific structure of the references on all four examples; the baselines either give generic descriptions, swap to a visually similar action, or recover only the coarse action label without the discriminative body part or temporal sequence.