Fine-grained Human Motion Understanding with Language Models

Thomas Markhorst*
Delft University of Technology

Zhi-Yi Lin
Delft University of Technology

Jouh Yeong Chew
Honda Research Institute Japan

Jan van Gemert
Delft University of Technology

Xucong Zhang
Delft University of Technology

Abstract

In this work, we propose FiGMo, an LLM-based model for fine-grained human motion understanding that represents motion as a sequence of skeletal poses with explicit timestamps for each pose. Each pose encodes body joint positions and is temporally grounded with timestamp tokens, allowing the model to reason about motion order, duration, and rhythm. To study what supervision is needed for motion-language reasoning, we construct a diverse training mixture spanning pose captioning, pose question answering, motion captioning, and motion question answering. Our ablations show that the primary gains come from the diversity of pose- and motion-level supervision, while staged training provides a smaller additional benefit. Different from previous works that rely on ground-truth 3D motion capture, our approach supports both 2D and 3D skeletal motion representations through a unified pose encoder, and can optionally incorporate video to provide contextual information. Extensive experiments on BABEL-QA, HuMMan-QA, CompMo, NTU-RGB+D, and QEVD-Coach demonstrate that our method achieves state-of-the-art performance across multiple benchmarks, highlighting the effectiveness of explicit temporal encoding and diverse pose- and motion-level supervision for fine-grained human motion understanding. Notably, even when using only 2D skeletal input, our approach surpasses previous 3D-based methods.

1 Introduction

Understanding human motion is a fundamental step toward building human-centered AI systems that can perceive, interpret, and interact with people in socially appropriate ways [69, 82, 44]. It supports a wide range of applications, including human-robot interaction [59], rehabilitation monitoring [58, 28], and sports coaching [60]. Despite substantial progress, achieving fine-grained and temporally grounded motion understanding remains challenging [11, 66, 60].

Recent advances in large language models (LLMs) have expanded multimodal reasoning across video understanding [39], motion captioning [70, 23], and behavior analysis [8]. Most of these approaches rely on video as the primary input modality. While video provides rich visual context, it requires processing high-dimensional pixel inputs that are computationally expensive to process and therefore typically require aggressive temporal subsampling [33, 76, 36]. This often limits temporal precision and makes it difficult to capture subtle, rapid, or fine-grained motion dynamics. Moreover, raw video

*t.c.markhorst@tudelft.nl

Preprint.

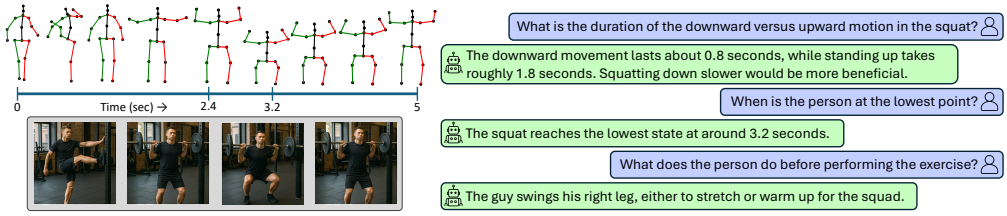


Figure 1: Given a motion sequence (top left), our method answers questions about fine-grained temporal aspects of human motion (right). By representing motion as timestamped skeletal poses, FiGMo supports reasoning about action order, timing, and duration.

inherently contains appearance information such as identity, clothing, and background context, which may introduce privacy concerns or demographic biases in downstream applications [28, 50, 6, 5]. Therefore, in many realistic deployment scenarios, video cannot be stored or transmitted due to bandwidth [61], regulatory [19], or privacy constraints [31], particularly in edge-device settings [2].

In contrast, representing humans as sequences of body joints provides a compact, explicit, and appearance-invariant description of motion [22, 17, 49, 83]. Such skeletal representations are significantly lower-dimensional than raw video and therefore enable high frame-rate processing while abstracting away sensitive visual attributes [22]. This makes skeleton-based reasoning particularly attractive for privacy-sensitive, fine-grained applications such as rehabilitation monitoring.

However, existing human motion-centric models often overlook detailed temporal reasoning. LLM-based models with VQ-VAE motion encoders [20, 38] compress entire motion clips into discrete tokens, losing pose-level timing. Similarly, a model [72] that directly feeds pose sequences into an LLM, concatenates poses without explicitly grounding their timestamps. Consequently, motion clips are treated as atomic units rather than temporally structured sequences, limiting the model's ability to reason about the exact timing and duration of motion.

In this work, we present FiGMo, an LLM-based method for **Fi**ne-**G**rained **Mo**tion understanding that explicitly models human motion as a temporally grounded sequence of skeletal poses. Each pose is associated with a timestamp, enabling structured reasoning over motion order and timing. Beyond the motion representation itself, we study what supervision is needed to train motion-language LLMs effectively. We construct a diverse pose-to-motion training mixture spanning pose captioning, pose question answering, motion captioning, and motion question answering. We compare training on the full mixture in a single stage with a staged pose-to-motion schedule, and our ablations show that the primary gains come from diverse pose- and motion-level supervision, while staged training provides a smaller additional benefit.

Unlike many prior motion-based methods that rely on clean 3D MoCap skeletons [20, 10, 21] captured in controlled environments [48], our approach supports both 2D and 3D skeleton inputs. We propose a pose encoder that learns consistent representations from mixed and partially corrupted skeletal data, improving robustness to noisy 2D-pose detections obtained from monocular videos. When available, our method can incorporate video to provide supplementary context.

We evaluate FiGMo using multiple human motion understanding benchmarks. The results demonstrate state-of-the-art performance across all evaluated datasets, highlighting the effectiveness of explicit timestamp grounding, unified 2D/3D pose encoding, and diverse pose- and motion-level supervision for motion-centric reasoning.

In summary, our contributions are:

- an LLM-based method, FiGMo, for fine-grained human motion understanding that explicitly models motion as a temporally grounded pose sequence;
- a unified pose encoder that supports both 2D and 3D skeletal inputs, enabling the same model to operate on practical 2D detections as well as 3D pose sequences;
- an analysis showing that pose-level supervision substantially improves motion understanding, while staged training adds smaller gains;
- state-of-the-art performance across multiple human motion understanding benchmarks, with a 2D-only variant outperforming prior 3D-based methods.

2 Related Work

2.1 Motion Representation

Early action recognition models focused on end-to-end video classification using convolutional and transformer-based architectures. Two-dimensional CNNs such as TRN [80] and TSM [41] captured short-term temporal cues through sparse frame sampling, while 3D CNNs, including C3D [65] and I3D [7] extended these ideas to dense spatio-temporal modeling. More recently, Video-LLMs have demonstrated strong performance in appearance-based understanding tasks [63, 74]. However, they operate on high-dimensional pixel inputs that often require aggressive temporal subsampling to remain computationally feasible [33, 76]. This can obscure subtle motion cues and temporal relationships. Furthermore, video contains appearance information such as identity and ethnicity, which may introduce privacy or deployment constraints in real-world applications [28, 50, 6, 5].

Skeleton-based approaches alleviate these issues by modeling motion as joint coordinate sequences rather than raw appearance. Graph-based and transformer-based architectures such as ST-GCN [73], PoseC3D [17], and MotionBERT [83] are robust to viewpoint changes and support high-frame-rate processing. MotionBERT further improves generalization through AMASS pretraining [48] with random joint masking and additive noise. However, these models remain confined to clip-level classification [7, 32, 62] without explicit reasoning about fine-grained motion timing [18].

2.2 Human Motion Understanding

To move beyond coarse skeletal action classification, several datasets have been introduced for more fine-grained motion understanding. HumanML3D [24], BABEL [54], CompMo [71] and Motion-X [42, 79] provide motion–language pairs with semantics and temporal annotations. BABEL-QA [18] and HuMMan-QA [34] go beyond captioning by posing questions about specific events and relationships within a motion sequence, enabling evaluation of reasoning over both the content and temporal dynamics of human motion.

Using these datasets, cross-modal alignment models such as MotionCLIP [64] learn shared motion–text embeddings for retrieval and semantic organization. This idea was extended by multimodal LLMs including MotionGPT [29], MotionGPT2 [68], and AvatarGPT [81], enabling text-to-motion synthesis, captioning, and motion-editing. DEMO [71] and UniMotion [35] move toward finer temporal modeling by segmenting and captioning motion sequences. Although capable of generating motion descriptions, these models perform captioning rather than answering questions about the content and temporal dynamics of the motion sequence.

Recent LLM-based models such as MotionLLM [10] and HuMoCon [20] extend earlier captioning approaches by training on motion question–answering datasets [18]. However, they largely retain the motion representations used for captioning. In particular, several works encode motion using discrete VQ-VAE tokens [20, 10, 68], which compress entire motion sequences into latent motion codes. This compression removes the explicit correspondence between individual poses and their temporal origin, preventing precise timestamp conditioning and limiting fine-grained temporal modeling. While LLaMo [38] represents motion as a sequence of skeletal poses rather than a compressed unit, it does not explicitly encode the temporal origin of each pose and instead focuses on key pose frames, making the temporal intervals between poses ambiguous. In contrast, our method represents motion as a sequence of skeletal poses with explicit timestamps, allowing the model to reason about motion timing at the pose level.

Another limitation of existing motion–LLM systems [10, 20, 38] is their reliance on clean 3D motion capture data collected in controlled environments [48]. This restricts their applicability in real-world settings where motion is often obtained from noisy 2D pose detectors applied to monocular RGB videos. Therefore, our model applies a pose encoder pre-trained for both 2D and 3D poses.

Moreover, recent motion-LLM methods primarily adapt LLMs with motion-level QA supervision [20, 10, 38]. In contrast, several video-LLM training recipes first use image-language data or image-based instruction tuning before incorporating video-language data [33, 76]. This suggests an analogous question for skeletal motion: whether pose-level supervision can support downstream motion understanding, and how it should be combined with motion-level supervision. We therefore construct a pose-to-motion supervision mixture and analyze how different components and training strategies affect skeletal motion-language reasoning.

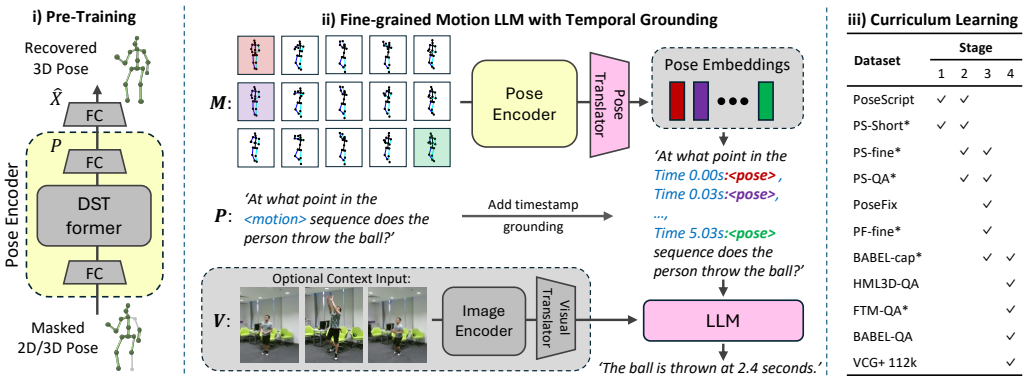


Figure 2: Pipeline of our FiGMo, comprising (i) pose encoder pre-training, (ii) encoding high-frequency motion M with explicit temporal grounding for fine-grained motion understanding, optionally with video input V for contextual cues, and (iii) pose- and motion-level supervision modules that can be used either in a single stage or through a staged pose-to-motion schedule. Datasets marked with * are proposed in this work.

3 Method

We develop an LLM-based method for fine-grained human motion understanding that models human movement as temporally grounded skeletal pose sequences. The model can interpret both 2D and 3D human motion represented as skeletal poses.

Each pose feature is grounded with an explicit timestamp, providing the LLM with access to the temporal origin of each body configuration. Beyond the motion representation itself, we construct a pose-to-motion supervision mixture that includes pose captioning, pose QA, motion captioning, and motion QA. This allows us to study how pose- and motion-level supervision contribute to skeletal motion-language reasoning, and how different training strategies affect performance. Fig. 2 provides an overview of the proposed model.

3.1 Unified Pose Encoder

To enable fine-grained 2D and 3D human motion understanding, we model human motion as a temporal sequence of static poses, where each pose is independently encoded to capture detailed spatial configurations of body joints. This design allows the model to focus on precise body structure at the frame level, while temporal reasoning is later handled by the LLM (see Sec. 3.3).

For pose encoding, we build upon the motion encoder of MotionBERT [83], originally developed for 2D-to-3D human pose lifting. Since our goal for the encoder is to represent individual poses rather than continuous motion, we retain only the spatial transformer blocks to encode body joints.

To make the encoder robust to noisy or incomplete inputs, we pre-train it to reconstruct 3D poses from corrupted 2D or 3D pose inputs. During training, we apply both additive noise and random joint masking [15, 4, 25], which regularizes the model and simulates real-world conditions such as occluded body parts or missing depth information. The encoder is trained using a mixture of detected 2D poses, 3D mocap data [48], and 2D projections of 3D mocap sequences, allowing it to generalize across pose representations.

During pre-training, the pose encoder maps each pose into a latent embedding P and reconstructs a 3D pose $\hat{X}$ following [83]. For mocap data with available 3D ground truth, we minimize a reconstruction loss $\mathcal{L}_{3D}$, while for 2D pose inputs, we use a re-projection loss $\mathcal{L}_{\text{rep}}$:

$$\mathcal{L}_{\text{3D}} = \sum_{j=1}^{J} \|\hat{X}_j - X_j\|_2, \quad \mathcal{L}_{\text{rep}} = \sum_{j=1}^{J} \delta_j \|\hat{x}_j - x_j\|_2, \tag{1}$$

with j for joint index, δ_j as joint visibility, X as 3D pose, x as 2D pose, and $\hat{}$ indicating reconstructed pose. This pretraining yields a unified and robust pose encoder capable of encoding both 2D and 3D skeletons consistently, providing a strong foundation for downstream fine-grained motion reasoning.

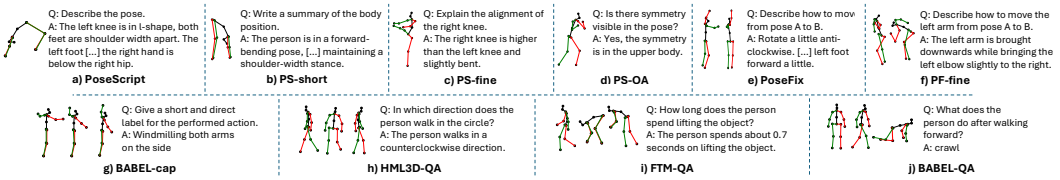


Figure 3: Training data examples. Pose-related datasets are shown in the top row and motion-related datasets in the bottom row. The modules provide complementary supervision, ranging from pose captions and pose QA to motion captions and temporally grounded motion QA. We analyze how these supervision components and training strategies affect skeletal motion-language reasoning.

3.2 Multimodal LLM Structure

While the pose encoder provides robust representations, its embeddings are not directly aligned with the semantic feature space of LLMs [10, 33, 39]. To bridge this modality gap, we introduce a lightweight projection module that maps pose embeddings into the LLM's embedding space. As shown in Fig. 2, this module is implemented as a two-layer multilayer perceptron (MLP), referred to as the *pose translator*. The translated features are then passed to the multimodal LLM $\mathcal{Q}$, which also accepts optional video inputs V through its visual encoder and text prompts P through its language interface. This unified architecture allows the LLM to combine motion, video, and text information while maintaining modularity. Unlike previous approaches [20, 38] that enforce tight alignment between motion and video modalities, we treat video as an auxiliary cue that enhances motion understanding, providing contextual grounding without constraining the motion representation space.

3.3 Interpreting Fine-grained Timing in Motion

The pose translator produces an embedding m that is aligned with the token space of the LLM $\mathcal{Q}$. This embedding can be inserted into a textual query, for example: *"In what position is this person <pose>?"*, where <pose> is replaced by the pose embedding m. The resulting prompt is then processed by $\mathcal{Q}$.

A naive extension from single-pose reasoning to motion reasoning would concatenate multiple pose embeddings, forming prompts such as *"What is this person doing <pose> ... <pose>?"*. However, this concatenation introduces two key limitations. First, it neglects the precise timing of each pose, making it difficult for the model to infer when specific movements occur within the sequence. Second, the temporal resolution of such an encoding is fixed, as there is no mechanism to indicate variations in motion frame rate. Consequently, motions that are temporally downsampled to fit within the LLM's context window, such as long sequences, are misinterpreted as faster or shorter actions.

To overcome these limitations, we introduce explicit timestamp conditioning. Each pose is concatenated with its corresponding timestamp, producing temporally grounded prompts such as: *"What is this person doing? Time: 0.00 s <pose>, Time: 0.03 s <pose>, ..., Time: 5.03 s <pose>."* Here, the token "*Time*" serves as a special indicator of pose timing, and the sampling interval can be flexibly adjusted. This explicit timestamping provides the LLM with fine-grained temporal structure, supporting reasoning about motion order, duration, and rhythm. Furthermore, it supports variable frame rates and adaptive temporal sampling strategies, allowing the model to process long motion sequences without losing temporal consistency. The shared temporal reference also facilitates reasoning over multi-person interactions by aligning poses from different individuals in time.

3.4 Training with Pose- and Motion-Level Supervision

We train the motion-language model using a diverse set of pose- and motion-level supervision modules derived from existing pose and motion datasets. These modules vary along two axes: the input type, ranging from isolated poses to full motion sequences, and the task type, ranging from captioning to question answering. This design lets us analyze how different supervision components contribute to skeletal motion-language reasoning, while also allowing the same data to be used either in a single training stage or in a staged schedule. Unlike recent motion-LLM methods [10, 20, 38], which primarily adapt LLMs with motion-level QA supervision, our training mixture also includes pose-level captioning and QA. Our supervision modules use standard sources also used in prior

motion-LLM work: AMASS [48], BABEL [54], and HumanML3D [24]. Thus, the main difference is not a larger data source, but reusing existing data at different granularities through pose- and motion-level supervision. Representative examples for all modules are shown in Fig. 3.

3.4.1 Pose Data Modules

We take samples from two human pose–text datasets: PoseScript [13] and PoseFix [14]. Both datasets use human poses from AMASS [48] and provide rule-based machine generated textual descriptions. To further enrich the prompt diversity of these datasets and enhance the understanding of human pose data, we propose multiple data augmentations, resulting in various data modules that we list below.

We re-annotate PoseScript and PoseFix using an open-source LLM $\mathcal{F}$ to increase linguistic and prompt diversity. For each pose, two captions are provided to $\mathcal{F}$ together with an instruction describing the desired output. Generated annotations are filtered with an LLM-as-a-judge consistency check against a held-out third caption, following prior re-annotation practices [67, 76]. This procedure produces PS-short, PS-fine, and PF-fine; further details are provided in the supplementary material.

PS-short. To provide concise pose-level supervision, we condense PoseScript's multi-sentence descriptions into one-sentence summaries. This module produces short, semantically clear captions.

PS-fine and PF-fine. Inspired by image QA datasets [75, 43], we generate fine-grained body-part QA pairs from PoseScript and PoseFix by instructing $\mathcal{F}$ to convert local limb and joint descriptions into questions and answers. These new modules link language to detailed spatial pose configurations, improving both structural interpretability and instruction-following ability.

PS-QA. We also generate structured QA pairs directly from 3D pose geometry, covering body symmetry, joint distances, joint angles, and foot positioning. Because the answers are computed from pose coordinates, this module provides automatically verifiable supervision for spatial body reasoning and for linking pose cues to language. We balance referenced body parts and answer distributions to improve coverage and avoid bias. Further implementation details are provided in Sec. B.1.2.

3.4.2 Motion Data Modules and Temporal Alignment

Following previous work [10, 20, 38], we prompt an LLM to generate complex QA pairs for motions sampled from HumanML3D [24] (HML3D-QA) and include samples from BABEL-QA in our training data. As these QA sources provide limited coverage of simple motion descriptions and temporally detailed questions, we introduce BABEL-cap and FTM-QA to complement them.

BABEL-cap. To introduce simple motion-level supervision, we construct this data module from AMASS [48] and BABEL [54]. Short clips extracted from AMASS are labeled with their corresponding BABEL actions to form concise question-answer pairs of the form *"What action is being performed?"*. This module complements HML3D-QA by providing simple motion-level captioning.

FTM-QA. We introduce Fine-grained Temporal Motion QA (FTM-QA) to supervise questions about action duration, start and end times, and temporal ordering. The module follows the LLM-based QA generation procedure of [10], originally applied to HumanML3D, but adapts it to BABEL's temporally dense action annotations. This produces motion-text pairs that explicitly support temporal alignment and fine-grained motion reasoning.

3.4.3 Training Strategies

We consider two training strategies for the same supervision mixture. In the single-stage strategy, all data modules are combined and used throughout training. In the staged strategy, the modules are introduced according to input and task type, moving from pose-level supervision to motion-level supervision. In Stage 1, we use PoseScript and PS-short to establish pose-language alignment through direct pose-caption correspondences. Stage 2 adds PS-fine and PS-QA to increase linguistic diversity and fine-grained body-part reasoning. Stage 3 combines pose-reasoning modules with BABEL-cap, introducing simple motion-level supervision. Finally, Stage 4 adds the complex motion-level datasets HML3D-QA, BABEL-QA, and FTM-QA for temporally detailed motion reasoning over longer sequences. Additionally, a small subset of VCG-plus 112k [47] reinforces video-language grounding.

Table 1: Comparison of our method, fine-tuned on BABEL-QA following previous work, with state-of-the-art methods on the BABEL-QA dataset. "Ours (3D)" achieves the best performance overall by using the 3D motion input. Remarkably, "Ours (2D)" still outperforms previous methods on most metrics, even though it uses only 2D motion, while all prior methods rely on 3D motion.

Model	Overall	Query Type			Temporal Filter			
		Action	Direction	Body Part	Before	After	In Between	Other
NSPose [18]	0.578	0.627	0.598	0.325	0.531	0.594	0.590	0.609
IMoRe II [34]	0.640	0.695	0.679	0.358	0.600	0.649	0.675	0.663
MotionLLM [10]	0.436	0.517	0.354	0.154	0.427	0.368	-	0.529
LLaMo [38]	0.458	0.525	0.398	0.224	0.443	0.392	-	0.518
HuMoCon [20]	0.711	0.809	0.697	0.623	0.707	0.635	-	0.797
FiGMo (2D)	0.750	**0.816**	0.681	0.550	0.677	0.758	**0.781**	**0.806**
FiGMo (3D)	**0.758**	0.791	**0.722**	**0.658**	**0.710**	**0.766**	0.750	0.797

The single-stage and staged strategies use the same data modules. Therefore, comparing them allows us to distinguish the effect of supervision diversity from the effect of staging. Unless otherwise specified, we report the staged variant as FiGMo because it performs best in our ablations.

4 Experiments

Training. The pose encoder is pre-trained following the MotionBERT [83] data pipeline, using 3D datasets Human3.6M [27] and AMASS [48], and 2D datasets PoseTrack [1] and InstaVariety [30]. During pre-training, 15% of joints are randomly masked, and additive noise sampled from a mixture of Gaussian and uniform distributions [9] is applied to enhance robustness.

We adopt VideoLLaMA3-7B [76] as the multimodal LLM $\mathcal{Q}$ and fine-tune the motion-language model using the pose- and motion-level supervision modules described in Sec. 3. Unless otherwise specified, FiGMo refers to the staged variant, which performs best in our ablations. The LLM $\mathcal{Q}$ is optimized using LoRA [26].

For 3D input, each pose is represented by joint coordinates (x, y, z). For 2D input, each joint is represented as (x, y, c), where c denotes detection confidence and is set to 1 for re-projected 3D data. Motion sequences are sampled at 30 fps and uniformly subsampled when exceeding 800 frames to maintain a consistent maximum sequence length.

Evaluation and Metrics. We report prediction accuracy on BABEL-QA and HuMMan-QA. For BABEL-QA, following [20, 10, 45], classifier-based methods are evaluated with direct accuracy, while generative methods are evaluated with an LLM-based answer-matching metric. This LLM-based evaluation is used only for BABEL-QA; HuMMan-QA accuracy is computed directly following [34]. In Sec. A.5, we analyze this LLM-evaluator and show that it closely matches direct answer evaluation. For CompMo [71], we report standard dense-captioning metrics and temporal localization metrics, including SODA, CIDEr, METEOR, tIoU, and F1. Additional experiments and qualitative assessments are provided in Sec. A.

4.1 Evaluation on BABEL-QA

We evaluate our method on the BABEL-QA benchmark to assess fine-grained motion understanding. All baselines use ground-truth 3D motion as input. We report results for both 3D motion input and 2D motion input, where the latter is obtained by reprojecting the 3D skeletons into the image plane using a pinhole camera model (see Sec. A.11), denoted as "Ours (3D)" and "Ours (2D)", respectively.

As shown in Tab. 1, FiGMo achieves the best overall performance. The gains are especially clear on temporally filtered questions such as "After" and "In Between", indicating that explicit timestamp conditioning helps the model reason about action order. Compared with prior motion-LLM methods that rely on compressed motion tokens or key poses, our timestamped pose representation preserves the temporal origin of each pose, which is important for fine-grained motion QA.

Table 2: Comparison of our method, fine-tuned on HuMMan-QA following previous work, with state-of-the-art methods on HuMMan-QA. Our method outperforms previous methods.

Model	Overall	Query Action				Query Direction			Query Body Part			
		All	Before	After	BTW	All	Before	After	All	Before	After	BTW
NSPose [18]	0.691	0.700	0.686	0.610	0.729	0.822	0.425	0.833	0.677	0.620	0.639	0.833
IMoRe I [34]	0.719	0.744	0.652	0.734	0.854	1.000	1.000	1.000	0.665	0.609	0.647	0.889
IMoRe II [34]	0.730	0.746	0.648	0.739	0.813	1.000	1.000	1.000	0.717	0.636	0.703	0.861
FiGMo (2D)	0.792	0.808	0.716	0.795	0.800	1.000	1.000	1.000	0.753	0.652	0.767	**1.000**
FiGMo (3D)	**0.824**	**0.841**	**0.770**	**0.828**	**0.900**	1.000	1.000	1.000	**0.782**	**0.696**	**0.781**	**1.000**

Remarkably, even with the less informative 2D motion input, our method outperforms all previous 3D-based approaches and achieves competitive performance with our 3D variant. Since both "Ours (2D)" and "Ours (3D)" are fine-tuned from the same FiGMo, this result underscores the strength of the unified pose encoder and training scheme, which generalize effectively across motion representations.

Additional experiments with detected 2D skeletons are provided in the appendix, including one-shot action recognition on NTU-RGB+D 120 (Sec. A.2) and exercise feedback on QEVD-Coach (Sec. A.3). These results further support the applicability of FiGMo beyond reprojected 2D poses. Experiments on NTU-RGB+D 120 also demonstrate the ability of our method to take video as supplementary input.

4.2 Evaluation on HuMMan-QA

To evaluate generalization beyond BABEL, we follow prior work [34] on the HuMMan-QA benchmark, which is built on the HuMMan-MoGen motion dataset [77]. We fine-tune our model on the HuMMan-QA training split following their proposed protocol [34]. We compare with strong classification-based baselines, NSPose [18], and IMoRe [34]. The generative motion-language models do not report on HuMMan-QA and lack public training code, and are therefore excluded.

As shown in Tab. 2, our method achieves the best performance across all evaluation metrics. These results indicate that the timestamped pose representation and pose- and motion-level supervision transfer beyond BABEL-QA, maintaining strong performance on a different motion distribution.

4.3 Evaluation on Dense Motion Captioning

To directly evaluate temporal grounding, we further test FiGMo on the CompMo dense motion captioning benchmark [71]. Unlike BABEL-QA and HuMMan-QA, which evaluate motion understanding through question answering, CompMo requires the model to generate temporally localized captions for motion segments. This setting evaluates both semantic motion understanding and temporal localization, making it a direct test of whether timestamped pose representations support fine-grained grounding in time.

As shown in Tab. 3, FiGMo substantially outperforms UniMotion [35] and DEMO [71] across all captioning and temporal metrics. The large gains in captioning metrics, including CIDEr, METEOR, ROUGE-L, and BLEU, show that the model produces more accurate motion descriptions. At the same time, the improvements in tIoU and F1 indicate much stronger temporal localization. These results show that explicit timestamp conditioning not only improves motion QA, but also enables accurate localization and description of fine-grained motion events. Qualitative comparisons on the original DEMO examples are provided in Sec. A.1, showing that FiGMo improves temporal localization and often produces more specific motion descriptions.

4.4 Component Ablations

We conduct ablations on BABEL-QA to analyze timestamp grounding, pose-level supervision, motion-level supervision, and the effect of staging the same supervision modules. All variants use 3D motion input and are fine-tuned on BABEL-QA. The results are summarized in Tab. 4.

Table 3: Comparison on the CompMo Dense Motion Captioning benchmark [71]. We report captioning metrics and temporal localization metrics, and outperform previous work on both captioning and temporal localization performance.

Model	Dense Captioning ↑							Temporal ↑	
	SODA	SODA-B	CIDEr	METEOR	ROUGE-L	BLEU@1	BLEU@4	tIoU	F1
UniMotion [35]	0.61	12.81	1.01	0.43	0.85	0.78	0.00	36.14	4.00
DEMO [71]	17.85	64.40	134.44	16.41	24.05	23.90	11.00	77.94	58.21
FiGMo	**50.98**	**91.00**	**456.95**	**38.77**	**60.48**	**58.41**	**45.38**	**98.44**	**97.48**

Table 4: Ablation study on BABEL-QA using 3D motion input. Timestamp grounding is important for temporal reasoning, while diverse pose- and motion-level supervision drives the main gains; staged training adds a smaller benefit.

Model	Overall	Query Type			Temporal Filter			
		Action	Direction	Body Part	Before	After	In Between	Other
w/o TS Ground	0.711	0.757	0.625	0.617	0.621	0.702	0.734	0.797
BABEL-QA only	0.443	0.481	0.389	0.342	0.383	0.435	0.594	0.500
w/o Pose Sup.	0.720	0.764	0.667	0.592	0.673	0.746	0.734	0.738
w/o Motion Sup.	0.724	0.797	0.597	0.558	0.665	0.742	0.781	0.759
Single-stage	0.751	0.793	0.708	0.617	0.698	0.758	0.828	0.790
Complete staged model	0.758	0.791	0.722	0.658	0.710	0.766	0.750	0.797

Removing timestamp grounding decreases overall accuracy from 0.758 to 0.711. The degradation is most visible for temporally sensitive query types such as "Before", "After", and "In Between", while queries without an explicit temporal filter ("Other") remain relatively stable.

Removing pose-level or additional motion-level supervision also hurts performance, reaching 0.720 and 0.724, respectively. The pose ablation is particularly informative: although BABEL-QA evaluates motion QA, removing static pose supervision substantially reduces performance, suggesting that pose-level supervision is important for downstream motion understanding. The motion ablation further shows that BABEL-QA fine-tuning captures part of the required reasoning, but broader motion supervision from our motion modules improves detailed understanding. When removing both pose- and motion-level supervision using only BABEL-QA fine-tuning, performance drops to 0.443, further confirming that the broader supervision mixture is essential.

Finally, to isolate the effect of staging from the effect of data composition, we train a single-stage variant using the same supervision modules as the complete model. This variant reaches 0.751 overall accuracy, close to the 0.758 obtained by the staged model. Thus, while the staged schedule performs best, the small gap indicates that the main performance driver is the diverse pose- and motion-level supervision rather than the staging itself.

5 Conclusion

We present FiGMo, an LLM-based model for fine-grained human motion understanding that represents motion as timestamped skeletal pose sequences. FiGMo combines a unified 2D/3D pose encoder with explicit timestamp conditioning, enabling reasoning over action order, duration, and temporally localized events. We further study pose- and motion-level supervision, showing that diverse supervision, especially pose-level data, is a key driver of downstream motion understanding, while staged training provides a smaller additional gain. Experiments on motion QA, dense motion captioning, and action recognition benchmarks demonstrate state-of-the-art performance, including strong results with only 2D skeletal input.

Future work could enrich timestamped motion-language reasoning with scene and object context. Skeletal poses capture body dynamics compactly, but omit information about objects, contact, and 3D scene layout that may be important for interaction understanding. Adaptive temporal sampling is also a promising direction for scaling to long sequences while preserving short but important actions.

References

[1] Mykhaylo Andriluka, Umar Iqbal, Eldar Insafutdinov, Leonid Pishchulin, Anton Milan, Juergen Gall, and Bernt Schiele. Posetrack: A benchmark for human pose estimation and tracking, 2018.

[2] Francesco Cosimo Andriulo, Marco Fiore, Marina Mongiello, Emanuele Traversa, and Vera Zizzo. Edge computing and cloud computing for internet of things: A review. *Informatics*, 11(4), 2024.

[3] Satanjeev Banerjee and Alon Lavie. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In Jade Goldstein, Alon Lavie, Chin-Yew Lin, and Clare Voss, editors, *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan, June 2005. Association for Computational Linguistics.

[4] Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. Beit: Bert pre-training of image transformers, 2022.

[5] A. Bălţăreţu, P. Benschop, and Jan C Van Gemert. Are we smply biased: Identifying ethical biases in action recognition. Master's thesis, Delft University of Technology, Delft, The Netherlands, 2025.

[6] Joy Buolamwini and Timnit Gebru. Gender shades: Intersectional accuracy disparities in commercial gender classification. In Sorelle A. Friedler and Christo Wilson, editors, *Proceedings of the 1st Conference on Fairness, Accountability and Transparency*, volume 81 of *Proceedings of Machine Learning Research*, pages 77–91. PMLR, 23–24 Feb 2018.

[7] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, pages 6299–6308, 2017.

[8] Szeyi Chan, Shihan Fu, Jiachen Li, Bingsheng Yao, Smit Desai, Mirjana Prpa, and Dakuo Wang. Human and llm-based voice assistant interaction: An analytical framework for user verbal and nonverbal behaviors. *arXiv preprint arXiv:2408.16465*, 2024.

[9] Ju Yong Chang, Gyeongsik Moon, and Kyoung Mu Lee. Poselifter: Absolute 3d human pose lifting network from a single noisy 2d human pose, 2020.

[10] 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. *arXiv preprint arXiv:2405.20340*, 2024.

[11] Xin Chen, Anqi Pang, Wei Yang, Yuexin Ma, Lan Xu, and Jingyi Yu. Sportscap: Monocular 3d human motion capture and fine-grained understanding in challenging sports videos. *IJCV*, 129(10):2846–2864, 2021.

[12] MMPose Contributors. Openmmlab pose estimation toolbox and benchmark, 2020.

[13] Ginger Delmas, Philippe Weinzaepfel, Thomas Lucas, Francesc Moreno-Noguer, and Grégory Rogez. Posescript: Linking 3d human poses and natural language. *IEEE TPAMI*, 2024.

[14] Ginger Delmas, Philippe Weinzaepfel, Francesc Moreno-Noguer, and Grégory Rogez. Posefix: Correcting 3d human poses with natural language, 2024.

[15] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

[16] Haodong Duan, Mingze Xu, Bing Shuai, Davide Modolo, Zhuowen Tu, Joseph Tighe, and Alessandro Bergamo. Skeletr: Towards skeleton-based action recognition in the wild. In *ICCV*, pages 13634–13644, 2023.

[17] Haodong Duan, Yue Zhao, Kai Chen, Dahua Lin, and Bo Dai. Revisiting skeleton-based action recognition. In *CVPR*, pages 2969–2978, 2022.

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

[19] European Parliament and Council of the European Union. Regulation (EU) 2016/679 of the European Parliament and of the Council.

[20] Qihang Fang, Chengcheng Tang, Bugra Tekin, Shugao Ma, and Yanchao Yang. Humocon: Concept discovery for human motion understanding. In *CVPR*, pages 7179–7190, 2025.

[21] Yao Feng, Jing Lin, Sai Kumar Dwivedi, Yu Sun, Priyanka Patel, and Michael J Black. Chatpose: Chatting about 3d human pose. In *CVPR*, pages 2093–2103, 2024.

[22] Xuesong Gao, Keqiu Li, Xiulong Liu, Jie Nie, Weiqiang Chen, and Yonghong Tian. Privacy-preserving 3-d skeleton-based video action recognition via graph convolution network. *IEEE Transactions on Consumer Electronics*, 71(2):6627–6641, 2025.

[23] Chuan Guo, Yuxuan Mu, Muhammad Gohar Javed, Sen Wang, and Li Cheng. Momask: Generative masked modeling of 3d human motions. In *CVPR*, pages 1900–1910, 2024.

[24] 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 (CVPR)*, pages 5152–5161, June 2022.

[25] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross B. Girshick. Masked autoencoders are scalable vision learners. *CoRR*, abs/2111.06377, 2021.

[26] 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, 2021.

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

[28] Ankit Jain, Rajendra Akerkar, and Abhishek Srivastava. Privacy-Preserving Human Activity Recognition System for Assisted Living Environments . *IEEE Transactions on Artificial Intelligence*, 5(05):2342–2357, May 2024.

[29] Biao Jiang, Xin Chen, Wen Liu, Jingyi Yu, Gang Yu, and Tao Chen. Motiongpt: Human motion as a foreign language. *NeurIPS*, 36:20067–20079, 2023.

[30] Angjoo Kanazawa, Jason Y. Zhang, Panna Felsen, and Jitendra Malik. Learning 3d human dynamics from video, 2019.

[31] Katrina Karkazis and Jennifer R. Fishman. Tracking u.s. professional athletes: The ethics of biometric technologies. *The American Journal of Bioethics*, 17(1):45–60, 2017. PMID: 27996918.

[32] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. Hmdb: A large video database for human motion recognition. In *2011 International Conference on Computer Vision*, pages 2556–2563, 2011.

[33] Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. Llava-onevision: Easy visual task transfer, 2024.

[34] 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, 2025.

[35] Chuqiao Li, Julian Chibane, Yannan He, Naama Pearl, Andreas Geiger, and Gerard Pons-moll. Unimotion: Unifying 3d human motion synthesis and understanding, 2024.

[36] KunChang Li, Yinan He, Yi Wang, Yizhuo Li, Wenhai Wang, Ping Luo, Yali Wang, Limin Wang, and Yu Qiao. Videochat: Chat-centric video understanding, 2024.

[37] Lei Li, Sen Jia, Jianhao Wang, Zhaochong An, Jiaang Li, Jenq-Neng Hwang, and Serge Belongie. Chatmotion: A multimodal multi-agent for human motion analysis. *arXiv preprint arXiv:2502.18180*, 2025.

[38] Lei Li, Sen Jia, Jianhao Wang, Zhongyu Jiang, Feng Zhou, Ju Dai, Tianfang Zhang, Zongkai Wu, and Jenq-Neng Hwang. Human motion instruction tuning. In *CVPR*, pages 17582–17591, 2025.

[39] Bin Lin, Yang Ye, Bin Zhu, Jiaxi Cui, Munan Ning, Peng Jin, and Li Yuan. Video-llava: Learning united visual representation by alignment before projection. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 5971–5984, 2024.

[40] Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics.

[41] Ji Lin, Chuang Gan, and Song Han. Tsm: Temporal shift module for efficient video understanding. In *ICCV*, pages 7083–7093, 2019.

[42] Jing Lin, Ailing Zeng, Shunlin Lu, Yuanhao Cai, Ruimao Zhang, Haoqian Wang, and Lei Zhang. Motion-x: A large-scale 3d expressive whole-body human motion dataset. *Advances in Neural Information Processing Systems*, 2023.

[43] Weifeng Lin, Xinyu Wei, Ruichuan An, Peng Gao, Bocheng Zou, Yulin Luo, Siyuan Huang, Shanghang Zhang, and Hongsheng Li. Draw-and-understand: Leveraging visual prompts to enable mllms to comprehend what you want, 2025.

[44] Zhi-Yi Lin, Thomas Markhorst, Jouh Yeong Chew, and Xucong Zhang. Polyslgen: Online multimodal speaking-listening reaction generation in polyadic interaction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 29379–29390, June 2026.

[45] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning, 2023.

[46] Jun Liu, Amir Shahroudy, Mauricio Perez, Gang Wang, Ling-Yu Duan, and Alex C Kot. Ntu rgb+ d 120: A large-scale benchmark for 3d human activity understanding. *IEEE TPAMI*, 42(10):2684–2701, 2019.

[47] Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Khan. Videogpt+: Integrating image and video encoders for enhanced video understanding, 2024.

[48] Naureen Mahmood, Nima Ghorbani, Nikolaus F Troje, Gerard Pons-Moll, and Michael J Black. Amass: Archive of motion capture as surface shapes. In *ICCV*, pages 5442–5451, 2019.

[49] Thomas Markhorst, Zhi-Yi Lin, Jouh Yeong Chew, Jan Van Gemert, and Xucong Zhang. Muppet: Multi-person 2d-to-3d pose lifting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 5320–5330, June 2026.

[50] Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. A survey on bias and fairness in machine learning, 2022.

[51] Raphael Memmesheimer, Simon Häring, Nick Theisen, and Dietrich Paulus. Skeleton-dml: Deep metric learning for skeleton-based one-shot action recognition. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3702–3710, 2022.

[52] Raphael Memmesheimer, Nick Theisen, and Dietrich Paulus. Sl-dml: Signal level deep metric learning for multimodal one-shot action recognition. In *2020 25th International conference on pattern recognition (ICPR)*, pages 4573–4580. IEEE, 2021.

[53] Sunny Panchal, Apratim Bhattacharyya, Guillaume Berger, Antoine Mercier, Cornelius Bohm, Florian Dietrichkeit, Reza Pourreza, Xuanlin Li, Pulkit Madan, Mingu Lee, Mark Todorovich, Ingo Bax, and Roland Memisevic. What to say and when to say it: Live fitness coaching as a testbed for situated interaction, 2024.

[54] Abhinanda R Punnakkal, Arjun Chandrasekaran, Nikos Athanasiou, Alejandra Quiros-Ramirez, and Michael J Black. Babel: Bodies, action and behavior with english labels. In *CVPR*, pages 722–731, 2021.

[55] Haoxuan Qu, Yujun Cai, and Jun Liu. Llms are good action recognizers, 2024.

[56] Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025.

[57] Alberto Sabater, Laura Santos, Jose Santos-Victor, Alexandre Bernardino, Luis Montesano, and Ana C Murillo. One-shot action recognition in challenging therapy scenarios. In *CVPR*, pages 2777–2785, 2021.

[58] Iosune Salinas-Bueno, Maria Francesca Roig-Maimó, Pau Martínez-Bueso, Katia San-Sebastián-Fernández, Javier Varona, and Ramon Mas-Sansó. Camera-based monitoring of neck movements for cervical rehabilitation mobile applications. *Sensors*, 21(6), 2021.

[59] Alessandra Sciutti, Martina Mara, Vincenzo Tagliasco, and Giulio Sandini. Humanizing human-robot interaction: On the importance of mutual understanding. *IEEE Technology and Society Magazine*, 37(1):22–29, 2018.

[60] Dian Shao, Yue Zhao, Bo Dai, and Dahua Lin. Finegym: A hierarchical video dataset for fine-grained action understanding. In *CVPR*, pages 2616–2625, 2020.

[61] Sachchidanand Singh. Optimize cloud computations using edge computing. In *2017 International Conference on Big Data, IoT and Data Science (BID)*, pages 49–53, 2017.

[62] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild, 2012.

[63] Yolo Y. Tang, Jing Bi, Siting Xu, Luchuan Song, Susan Liang, Teng Wang, Daoan Zhang, Jie An, Jingyang Lin, Rongyi Zhu, Ali Vosoughi, Chao Huang, Zeliang Zhang, Pinxin Liu, Mingqian Feng, Feng Zheng, Jianguo Zhang, Ping Luo, Jiebo Luo, and Chenliang Xu. Video understanding with large language models: A survey, 2025.

[64] Guy Tevet, Brian Gordon, Amir Hertz, Amit H Bermano, and Daniel Cohen-Or. Motionclip: Exposing human motion generation to clip space. In *ECCV*, pages 358–374. Springer, 2022.

[65] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*, pages 4489–4497, 2015.

[66] Yin Wang, Zhiying Leng, Frederick WB Li, Shun-Cheng Wu, and Xiaohui Liang. Fg-t2m: Fine-grained text-driven human motion generation via diffusion model. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 22035–22044, 2023.

[67] Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language models with self-generated instructions, 2023.

[68] Yuan Wang, Di Huang, Yaqi Zhang, Wanli Ouyang, Jile Jiao, Xuetao Feng, Yan Zhou, Pengfei Wan, Shixiang Tang, and Dan Xu. Motiongpt-2: A general-purpose motion-language model for motion generation and understanding. *arXiv preprint arXiv:2410.21747*, 2024.

[69] Philine Witzig, Rares Constantin, Nikola Kovacevic, and Rafael Wampfler. Multimodal dialog act classification for digital character conversations. In *Proceedings of the 6th ACM conference on conversational user interfaces*, pages 1–14, 2024.

[70] Qi Wu, Yubo Zhao, Yifan Wang, Xinhang Liu, Yu-Wing Tai, and Chi-Keung Tang. Motion-agent: A conversational framework for human motion generation with llms. 2025.

[71] Shiyao Xu, Benedetta Liberatori, Gül Varol, and Paolo Rota. Dense motion captioning, 2025.

[72] Sheng Yan, Yong Wang, Xin Du, Hongchang Jin, and Mengyuan Liu. Improving fine-grained understanding for retrieval in human motion and text. *IEEE Signal Processing Letters*, 2024.

[73] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *AAAI*, volume 32, 2018.

[74] Antoine Yang, Arsha Nagrani, Paul Hongsuck Seo, Antoine Miech, Jordi Pont-Tuset, Ivan Laptev, Josef Sivic, and Cordelia Schmid. Vid2seq: Large-scale pretraining of a visual language model for dense video captioning, 2023.

[75] Yuqian Yuan, Wentong Li, Jian Liu, Dongqi Tang, Xinjie Luo, Chi Qin, Lei Zhang, and Jianke Zhu. Osprey: Pixel understanding with visual instruction tuning, 2025.

[76] Boqiang Zhang, Kehan Li, Zesen Cheng, Zhiqiang Hu, Yuqian Yuan, Guanzheng Chen, Sicong Leng, Yuming Jiang, Hang Zhang, Xin Li, Peng Jin, Wenqi Zhang, Fan Wang, Lidong Bing, and Deli Zhao. Videollama 3: Frontier multimodal foundation models for image and video understanding, 2025.

[77] Mingyuan Zhang, Huirong Li, Zhongang Cai, Jiawei Ren, Lei Yang, and Ziwei Liu. Finemogen: Fine-grained spatio-temporal motion generation and editing, 2023.

[78] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert, 2020.

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

[80] Bolei Zhou, Alex Andonian, Aude Oliva, and Antonio Torralba. Temporal relational reasoning in videos. In *ECCV*, pages 803–818, 2018.

[81] Zixiang Zhou, Yu Wan, and Baoyuan Wang. Avatargpt: All-in-one framework for motion understanding planning generation and beyond. In *CVPR*, pages 1357–1366, 2024.

[82] Chen Zhu, Buzhen Huang, Zijing Wu, Binghui Zuo, and Yangang Wang. E-react: Towards emotionally controlled synthesis of human reactions. *arXiv preprint arXiv:2508.06093*, 2025.

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

Sec. A expands on the experiments and evaluation protocols from the main paper. Sec. B details the construction of the pose- and motion-level supervision modules. Finally, Secs. C–E discuss limitations, broader impact, compute resources, and assets used in this work.

A Extra Experiments, Training & Evaluation Details

This section provides additional experiments that complement the main paper, the evaluation details are described in this section but not listed here:

- **Qualitative Comparison on Dense Motion Captioning** (Sec. A.1). We qualitatively compare results with DEMO [71], showing that FiGMo performs better at both temporal localization and captioning.
- **Real detected 2D skeletons on NTU-RGB+D 120** (Sec. A.2). We evaluate FiGMo on one-shot action recognition using detected 2D poses, showing that the model transfers beyond clean or reprojected skeletons.
- **Downstream exercise feedback** (Sec. A.3). We test whether explicit skeletal motion representations help provide feedback on exercise execution from detected 2D poses.
- **Qualitative examples** (Sec. A.4). We show examples of timestamp-aware motion reasoning, action decomposition, and semantic pose interpretation.
- **BABEL-QA GPT-evaluation analysis** (Sec. A.5). We compare strict string matching with GPT-based semantic matching to verify that the LLM-based BABEL-QA evaluator reflects meaningful answer correctness.

A.1 Dense Motion Captioning (CompMo) Qualitative Comparison

Fig. 4 shows a qualitative comparison between FiGMo and DEMO [71] on CompMo [71]. To avoid cherry-picking examples in favor of FiGMo, we use the same qualitative examples selected in the original DEMO paper. The pose renderings are also reused from the original DEMO visualization; we only add the predictions of FiGMo for comparison.

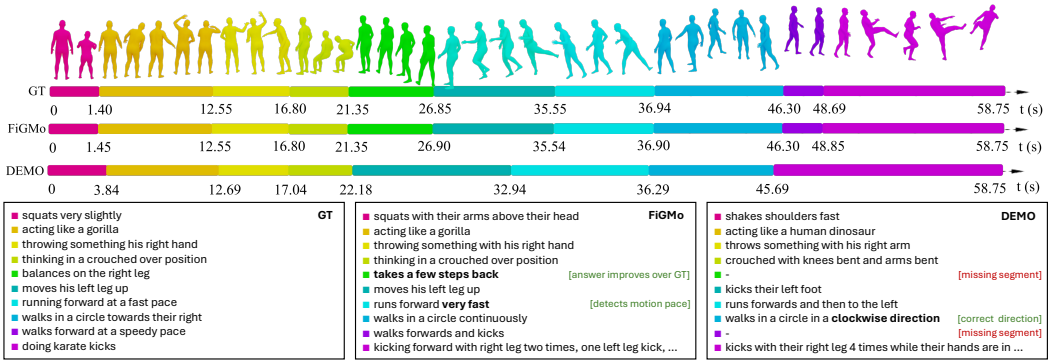
Consistent with the quantitative results in Tab. 3, FiGMo shows stronger temporal localization than DEMO. In these examples, FiGMo detects all annotated segments, while DEMO misses some segments, and the predicted temporal boundaries are generally better aligned with the ground truth. For captioning, DEMO already produces reasonable descriptions, but FiGMo captures several motion concepts more precisely. For example, in Fig. 4a, FiGMo captions ■ as "*takes a few steps back*", which appears more specific to the visualized motion than the ground-truth caption. Similarly, ■ is captioned as "*runs forwards very fast*", reflecting the fast pace described in the ground truth. Fig. 4b shows a similar pattern: for ■, FiGMo correctly identifies the swimming stroke as "*freestyle*", whereas DEMO does not.

A.2 Evaluation on NTU-RGB+D 120

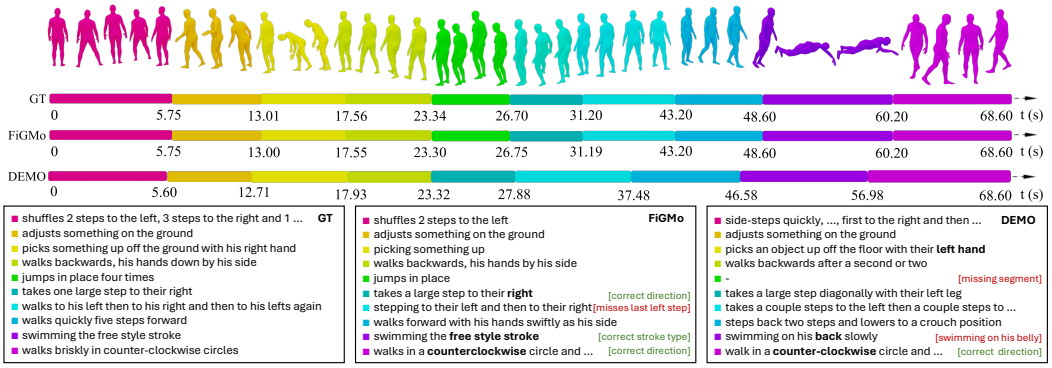
We further evaluate our model on the NTU-RGB+D 120 dataset under the one-shot recognition protocol [83, 46]. This one-shot setting is challenging, as i) the model must generalize to unseen actions from a single labeled instance, and ii) it also contains various two-person actions. Following prior work, we use 2D skeleton detections as motion input and additionally evaluate a multimodal variant that combines motion and video. The models are fine-tuned on an auxiliary set of 100 classes and one example per class from the 20 test categories. We report classification accuracy over the 20 classes in the one-shot evaluation set.

As shown in Tab. 5, our model achieves 69.2% accuracy using only motion input, surpassing all previous methods. When video is added as an auxiliary modality, performance rises to 77.4%, establishing a new state of the art on this benchmark. To our knowledge, this is the first evaluation of combined motion-video modeling in the one-shot setting of [46], suggesting that visual context can effectively complement motion features for action understanding.

To examine the role of video in the multimodal model, we also fine-tune and evaluate FiGMo using video input alone. This variant achieves 66.8% accuracy, which is lower than the motion-only model.



(a)



(b)

Figure 4: Qualitative comparison of FiGMo with DEMO [71] on the CompMo dense motion captioning dataset [71]. We use the original qualitative examples and pose renderings from the DEMO paper to avoid cherry-picking examples in favor of FiGMo. Across these examples, FiGMo shows stronger temporal localization and more precise captions.

This suggests that explicitly representing pose trajectories is beneficial compared with relying on sparsely sampled RGB frames alone.

Finally, since NTU includes two-person actions, this experiment also demonstrates that the model can process multi-person motion sequences in interaction-based activities.

Table 5: Comparison on the one-shot NTU-RGB+D 120 dataset using 2D skeleton detections.

Model	Acc.
ST-LSTM + AvgPool [16]	42.9
APSR [46]	45.3
TCN Oneshot [57]	46.5
SL-DML [52]	50.9
Skeleton-DML [51]	54.2
MotionBERT [83]	67.4
FiGMo (2D Motion)	69.2
FiGMo (2D Motion + Video)	**77.4**

A.3 Evaluation on Downstream Task

As an additional downstream evaluation, we test FiGMo on an exercise-feedback task derived from the QEVD-Coach dataset [53]. This dataset contains videos of people performing physical exercises

together with question-answer pairs describing exercise execution. For the downstream application test, we report language similarity metrics METEOR [3], ROUGE-L [40], and BERTScore [78].

From the original 300k samples, we select a subset of 30k examples and construct a train-test split. For the motion-based setting, 2D skeletons are extracted from the videos using MMPose [12]. All models are fine-tuned on the same data yet operate on different input modalities. The video and text-only training settings follow the protocols proposed in VideoLLaMA3 [76] and Qwen2.5 [56]. Note all compared methods use Qwen2.5 as the backbone model, ensuring that performance differences primarily stem from the input modality.

Tab. 6 shows the results. Our motion-based model outperforms VideoLLaMA3, which operates directly on raw video frames. The results suggest that explicit skeletal motion representations provide strong signals for evaluating exercise execution. Qualitatively, the video model struggles with subtle differences between visually similar exercises (e.g., planks versus push-ups), which are difficult to infer from temporally sparse sampled video frames. The text-only model operates without video or motion input and therefore provides a lower bound.

Table 6: Comparison on the downstream exercise-feedback task on QEVD-Coach. We compare FiGMo using detected 2D skeletal motion against a video LLM operating on raw video, and include a text-only LLM as a lower-bound baseline.

Model	Modality	METEOR	ROUGE-L	BERTScore
Qwen2.5 [56]	Text	0.859	0.849	0.978
VideoLLaMA3 [76]	Video	0.941	0.935	0.990
FiGMo	2D Motion	**0.946**	**0.940**	**0.991**

A.4 Qualitative Results

We present qualitative examples from our method to illustrate its capability in fine-grained temporal and semantic motion understanding. Fig. 5 shows representative cases of instructional queries on test samples. The left example demonstrates that the method not only identifies actions but also understands their temporal structure. For a dancing clip, the method correctly decomposes a complex movement into smaller sub-actions when prompted, showing that large composite motions can be parsed into meaningful fine-grained segments. The middle example shows that given a 60-second motion clip, the method accurately links a specific timestamp to a momentary action lasting less than a second. The right example illustrates that the method can associate the pose with relevant sports contexts (e.g., basketball or football), demonstrating its ability to ground pose semantics in broader human activity domains. Together, these qualitative results illustrate the strengths of our method in timestamp-aware motion reasoning, action decomposition, and semantic pose interpretation.

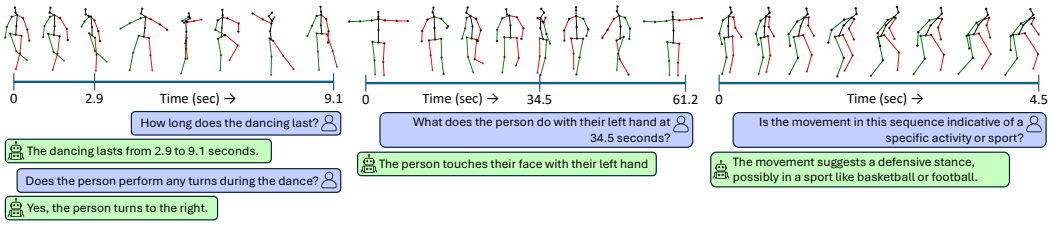


Figure 5: Qualitative examples showing fine-grained temporal motion reasoning (left), action decomposition from a minute-long complex motion sequence (middle), and semantic understanding of human activity (right).

A.5 BABEL-QA: Effect of GPT-3.5-turbo Evaluation

We compare strict string matching with a semantics-aware GPT-based evaluation (GPT-eval), which uses GPT-3.5-turbo to judge semantic equivalence between the model output and the ground truth. The evaluation follows the protocol used in prior work on BABEL-QA; further details are provided in Sec. A.8. Tab. 7 reports the results.

Semantic Equivalence Gains. GPT-eval consistently yields higher scores because it recognizes semantic similarity (e.g., *"jog"* vs. *"run"*). Our 2D model improves by $+4.9$ points overall ($0.707 \rightarrow 0.750$), especially in categories such as Action and Body Part, where minor lexical variations are common.

Model Superiority. Even under strict string matching—a harsher metric than used for the HuMoCon baseline—our 3D model achieves an Overall Score of 0.728, exceeding HuMoCon's GPT-evaluated score of 0.711. Under GPT-eval, our model reaches 0.758, establishing clear superiority across evaluation protocols.

Evaluation Robustness. To verify that GPT-eval does not inflate results indiscriminately, we examine the Direction category, which has a closed set of four mutually exclusive labels (right, left, forward, backward). Scores are identical under both metrics (2D: 0.681, 3D: 0.722), confirming that GPT-eval only relaxes evaluation when semantic interpretation is genuinely required.

Table 7: Comparison of strict string matching and GPT-3.5-turbo evaluation on BABEL-QA. HuMoCon uses GPT-eval. Our models outperform baselines under both evaluation schemes.

Model	Eval	Overall	Query Type			Temporal Filter			
			Action	Direction	Body Part	Before	After	In Between	Other
HuMoCon (3D)	GPT-eval	0.711	0.809	0.697	0.623	0.707	0.635	-	0.797
FiGMo (2D)	Str.-match	0.707	0.785	0.681	0.400	0.637	0.718	0.750	0.759
FiGMo (2D)	GPT-eval	0.750	0.816	0.681	0.550	0.677	0.758	0.781	0.806
FiGMo (3D)	Str.-match	0.728	0.770	0.722	0.550	0.677	0.742	0.719	0.759
FiGMo (3D)	GPT-eval	0.758	0.791	0.722	0.658	0.710	0.766	0.750	0.797

A.6 4-staged approach.

To further assess the performance of our 4-staged approach, we evaluate performance after each of the four training stages. Specifically, we report performance on BABEL-QA after each stage without additional fine-tuning: stage 1: 0.031, stage 2: 0.139, stage 3: 0.371, stage 4: 0.747, and after fine-tuning: 0.758. The results show a consistent performance increase throughout the curriculum, indicating that each stage contributes meaningful improvements. While the largest gain occurs in stage 4, earlier stages provide the necessary spatial and intermediate motion representations that enable this final jump in performance. This performance spike aligns with the objective of stage 4, which focuses on complex motion understanding, the capability evaluated by BABEL-QA.

A.7 Evaluation on ActivityNet-QA

For completeness, we evaluate our model on ActivityNet-QA, as prior motion-oriented works such as MotionLLM [10] and HuMoCon [20] report results on this benchmark using video-only input. ActivityNet-QA is a general-purpose video question answering dataset and is not specifically designed to assess detailed human motion reasoning. Only a small fraction of the questions (approximately 10%) are directly related to human motion, and an even smaller portion requires the fine-grained temporal understanding targeted by our approach.

On ActivityNet-QA, we follow [10] and use LLM-based evaluation. The results in Tab. 8 show that our method achieves comparable or slightly better performance than previous motion-oriented works, although our model is primarily designed for motion-based reasoning and the video is only an auxiliary modality. These results demonstrate that the proposed framework maintains strong generalization even when applied outside its primary data domain.

Table 8: Comparison on the ActivityNet-QA benchmark against prior motion-understanding LLMs. Although video is used only as an auxiliary input modality in our model, we achieve on-par or slightly better performance compared to methods designed for video-only inputs.

Model	Acc. (%) ↑	Score ↑
MotionLLM [10]	53.3	3.5
HuMoCon [20]	54.2	**3.6**
FiGMo	**54.4**	**3.6**

```
System Prompt
You are an intelligent chatbot designed for evaluating the correctness of
generative outputs for question-answer pairs.  Your task is to compare
the predicted answer with the correct answer and determine if they match
meaningfully.

Instructions:

      • Focus on the meaningful match between the predicted answer and the
        correct answer.
      • Consider synonyms or paraphrases as valid matches.
      • Evaluate the correctness of the prediction compared to the answer.

User Prompt
Please evaluate the following motion-based question-answer pair:
Question:  {question}
Correct Answer:  {answer}
Predicted Answer:  {pred}
Provide your evaluation only as a score between 0 and 1, where 1 indicates the
highest meaningful match.  Return the result strictly as a Python dictionary
string of the form {'score':  FLOAT}.
Do not add any explanation or additional text.

Examples:
What does the person do?
sit vs run -> {'score':  0.0}
sit vs sit -> {'score':  1.0}
sit vs sit down -> {'score':  1.0}
sit vs sit/walk -> {'score':  0.5}
run vs jog -> {'score':  1.0}
right hand vs right hand -> {'score':  1.0}
right hand vs right arm -> {'score':  0.5}
right hand vs left arm -> {'score':  0.0}
```

Figure 6: Prompt template used for GPT-based semantic evaluation on BABEL-QA.

A.8 GPT Evaluation Prompts

BABEL-QA Evaluation. To evaluate generative answers on BABEL-QA, we follow [10, 20, 38, 37] to use a semantics-aware GPT-based evaluation protocol. For each question–answer pair, we query a GPT model and ask it to judge whether the model's prediction is semantically consistent with the ground-truth answer. This allows synonyms, paraphrases, and linguistically natural variations to be considered correct when appropriate.

All evaluations use `gpt-3.5-turbo-0125`, following previous work. Queries are executed via the batch-completions API. For each evaluation sample, we prepare a pair of prompts: (i) a system prompt describing the evaluation rules, and (ii) a user prompt containing the detailed instruction, question, predicted answer, and ground truth. The GPT API returns one semantic similarity score per item, which we parse and aggregate into the final accuracy.

The GPT model is instructed to output only a Python dictionary string of the form: `{'score': <float>}`, where the score lies between 0 and 1. This strict output structure simplifies automatic parsing and metric computation. We show the full prompt template used for GPT-based scoring in Fig. 6.

ActivityNet-QA Evaluation. For ActivityNet-QA, we follow [20, 39] to evaluate generative answers using a GPT-based semantic scoring protocol. Similar to BABEL-QA, we query a GPT model to judge whether a predicted answer is semantically consistent with the ground truth, but we adopt the official ActivityNet-QA scoring scheme in which the model must output both a binary correctness flag and a discrete score in the range $[0, 5]$.

All experiments use `gpt-3.5-turbo-0125` through the batch-completions API. For each sample, we provide two messages: (i) a system prompt specifying the evaluation rules, and (ii) a user prompt containing the question, predicted answer, and ground truth. The GPT model returns a Python

System Prompt

```
You are an intelligent chatbot designed to evaluate the correctness of
generative outputs for question-answer pairs. Compare the predicted answer
with the correct answer and determine whether they match meaningfully.
Instructions:
```

- Focus on semantic correctness rather than literal matching.
- Accept synonyms and paraphrases when appropriate.
- Evaluate the predicted answer and assign a correctness flag and score.

User Prompt

```
Please evaluate the following video-based question-answer pair:
Question: {question}
Correct Answer: {answer}
Predicted Answer: {pred}
Return your evaluation strictly as a Python dictionary of the form:
{'pred': 'yes'/'no', 'score': INTEGER}
where score is an integer between 0 and 5.
Do not provide any additional explanation or text.
```

Figure 7: Prompt template used for GPT-based semantic evaluation on ActivityNet-QA.

dictionary containing `'pred'` ("yes"/"no") and `'score'` (integer in $[0, 5]$), which we parse and aggregate following the standard ActivityNet-QA evaluation. The full prompt used for GPT-based scoring is shown in Fig. 7.

A.9 One-Shot Training on NTU-RGB+D 120

For NTU-RGB+D 120, we follow the one-shot recognition protocol, where only a single labeled example is available for each of the 20 test classes. Because an LLM does not operate with a fixed classification vocabulary, we adapt the zero-shot finetuning procedure following the strategy proposed in prior LLM-based action recognition work [55].

During finetuning, we use the full auxiliary set together with the single example from each test class. For every training sample, we provide the model with a list of 20 randomly selected candidate class names that includes the correct class. The list order is randomly shuffled, and class names are paraphrased to prevent memorization of fixed strings. The model is instructed to predict the index of the correct class in the provided list, along with the corresponding paraphrased class name.

At test time, we supply only the 20 true test classes, again in a randomly shuffled order. The model must identify the correct class solely from this list, without exposure to the unseen class names outside their single example. This setup ensures a controlled one-shot evaluation while enabling the LLM to perform classification despite having no closed output vocabulary. Moreover, it is a fair comparison against existing baselines.

A.10 Training Parameters

This section summarizes the implementation details used across the curriculum training stages. All stages use a cosine learning-rate scheduler with a warm-up ratio of 0.03. Following Video-LLaMA3, we set the maximum token length to 16,384. In Stage 1, the Pose Encoder is initialized with the pretrained robust Pose Encoder from Sec. 3.1, the Pose Translator is randomly initialized, and the LLM is initialized from Video-LLaMA3-7B. Only the Pose Encoder and Pose Translator are optimized in this stage to align them with the LLM feature space; the LLM remains frozen. For Stages 2–4, we initialize from the previous stage and jointly optimize all components, including the LLM. Once LLM training begins, the learning rates for all modules are reduced. We finetune the LLM using LoRA with rank 64. The global batch size is 128. Motion data is processed at 30 fps with a maximum sequence length of 800 frames. Longer sequences are uniformly downsampled to satisfy this constraint. Tab. 9 lists the learning rates for each module across training stages, including the optional finetuning step.

Table 9: Learning rates of the modules in FiGMo across the four-stage curriculum and optional finetuning. Stage 1 aligns the Pose Encoder and Translator to the LLM feature space; later stages optimize the full model.

Stage	Model		
	Pose Encoder	Pose Translator	LLM
1	1.0×10^{-5}	1.0×10^{-3}	frozen
2	2.0×10^{-6}	1.0×10^{-5}	1.0×10^{-5}
3	2.0×10^{-6}	1.0×10^{-5}	1.0×10^{-5}
4	2.0×10^{-6}	1.0×10^{-5}	1.0×10^{-5}
(finetune)	2.0×10^{-6}	1.0×10^{-5}	1.0×10^{-5}

A.11 Pinhole Projection Details.

For experiments with reprojected 2D input, we convert each 3D motion sequence $\mathbf{P} \in \mathbb{R}^{T \times J \times 3}$ into a 2D pose sequence using a fixed pinhole camera. For each joint $\mathbf{p}_{t,j} = [X_{t,j}, Y_{t,j}, Z_{t,j}]^\top$, we first apply a rotation around the x-axis with tilt angle $\theta = 15^\circ$:

$$\mathbf{R}_x(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix}, \qquad \tilde{\mathbf{p}}_{t,j} = \mathbf{R}_x(\theta)\mathbf{p}_{t,j}.$$

We then translate the rotated pose in depth by a fixed offset $d = 4.0$ and clamp the depth for numerical stability:

$$\tilde{Z}_{t,j} \leftarrow \max(\tilde{Z}_{t,j} + d, 10^{-3}).$$

Using focal length $f = 5.0$ and principal point $(c_x, c_y) = (0, 0)$, each joint is projected as

$$u_{t,j} = f\frac{\tilde{X}_{t,j}}{\tilde{Z}_{t,j}} + c_x, \qquad v_{t,j} = f\frac{\tilde{Y}_{t,j}}{\tilde{Z}_{t,j}} + c_y.$$

Finally, we apply a fixed affine normalization to the projected coordinates:

$$\begin{bmatrix} u_{t,j} \\ v_{t,j} \end{bmatrix} \leftarrow 0.7 \left(\begin{bmatrix} u_{t,j} \\ v_{t,j} \end{bmatrix} + \begin{bmatrix} 0.03 \\ -0.07 \end{bmatrix} \right).$$

The final 2D representation stores the projected coordinates together with a constant confidence channel,

$$\mathbf{q}_{t,j} = [u_{t,j}, v_{t,j}, 1]^\top,$$

resulting in $\mathbf{Q} \in \mathbb{R}^{T \times J \times 3}$. The third channel is therefore a confidence indicator and does not encode depth. All samples use the same synthetic camera parameters rather than sequence-specific camera calibration.

B Data Generation

In this section, we further detail the generation of the data modules we propose in the paper. Tab. 10 gives an overview of the different data modules, where in our curriculum they are used, and how many samples each module contains. Sec. B.1 further details the pose curriculum, while Sec. B.2 details the motion curriculum.

B.1 Pose Curriculum

B.1.1 LLM Re-Annotation & Judging

PS-Short. To construct PS-Short, we convert the long PoseScript [13] descriptions into concise one-sentence summaries. For each pose, we use Qwen2.5-VL-7B-Instruct, conditioning the model on both the rendered SMPL image and two of the original PoseScript descriptions. The model produces a short and accurate summary without referencing the image or hallucinating additional details.

Table 10: Overview of the usage of different data modules over the stages in our curriculum. Additionally, we show the number of samples in each data module. *Indicates we propose the data module ourselves.

Dataset / Module	Num. Samples	Stages			
		1	2	3	4
PoseScript [13]	210k	✓	✓		
PS-Short*	70k	✓	✓		
PS-fine*	280k		✓	✓	
PS-QA*	30k		✓	✓	
PoseFix [14]	94k			✓	
PF-fine*	181k			✓	
BABEL-cap*	27k			✓	✓
HML3D-QA [10, 24]	32k				✓
FTM-QA*	64k				✓
BABEL-QA [18]	2k				✓
VCG+ 112k [39]	23k				✓

To ensure quality, every generated summary is then judged using a separate prompt. The same model receives the ground-truth description together with multiple candidate summaries (one correct, several distractors) and selects which one best matches the pose semantics. Only summaries judged correct are retained.

Fig. 8 shows the exact prompt templates used for both generation and judging.

Generation Prompt

```
Describe the human pose in this image accurately and concisely in one
sentence.  Use the following two extensive descriptions of the pose to help
you understand it better, but do not include any extra information that is
not in the descriptions.  Do not mention anything about SMPL or image in the
summary.

Image:  <SMPL .png render>
Descriptions:
1.  <desc_1>
2.  <desc_2>
```

```
Expected output:
<A concise one-sentence summary of the pose.>
```

Judging Prompt

```
You receive a description of a human pose or activity.
Below are five summaries of the description, one of which is the correct
summary.  Only respond with the corresponding multiple-choice letter.

 Description:  <desc_3>
A) {summary_1}
B) {summary_2}
C) {summary_3}
D) {summary_4}
E) {summary_5}
```

```
Expected output:  C
```

Figure 8: Prompt templates used to construct PS-Short. Top: generation prompt used by Qwen2.5-VL to produce concise one-sentence summaries from SMPL renders and PoseScript descriptions. Bottom: judging prompt used to automatically filter incorrect summaries.

PS-fine. To construct PS-fine, we target fine-grained pose understanding by extracting body-part–level motion descriptions from the original PoseScript [13] data. Unlike PS-short, which

summarizes an entire pose, PS-fine captures *local* motion semantics: for each pose transition description, we ask LLaMA-3.3-70B-Instruct to (i) select two body parts mentioned in the description, (ii) generate a short and precise sentence describing the movement of each selected body part, and (iii) produce an 'opposite' version of the pose by minimally altering attributes such as location or configuration. These structured JSON outputs yield compact, fine-grained pose primitives suitable for training the LLM on localized pose reasoning.

As in PS-Short, all generated descriptions undergo automatic quality control. For each pose sample, we construct a multiple-choice prompt with four options: constructed out of the correct and 'opposite' descriptions of the two body parts. The LLM receives another full PoseScript description together with the four candidate options and selects the correct one. Only body-part descriptions passing this verification step are retained. Fig. 9 presents the prompt templates used for PS-Fine generation and automatic judging.

Generation Prompt

```
You are given a description of a pose:
<desc_1>

Task:
    • Pick two body parts that are described in the description.
    • Describe the pose for the chosen body part, always include the body
      part in the sentence but don't describe other body parts.
    • Describe the opposite of the pose, only alter the specific difference
      (i.e. higher instead of lower, straight instead of bent).

Respond only with two JSONs. Do not include explanations, reasoning, or
repeat the prompt.
Output format:
{"body_part": "<chosen body part>",
"description": "<one short sentence>",
"opposite": "<one short sentence>"}
{"body_part": "<chosen body part>",
"description": "<one short sentence>",
"opposite": "<one short sentence>"}
```

Judging Prompt

```
You are given a description of a pose:
<desc_2>

Below are four pose descriptions of <body_part1> and <body_part2>. Pick the
correct option. Only return a JSON dictionary of the form:
{"choice": "a"}

A)<body_part1 correct> <body_part2 correct>
B)<body_part1 correct> <body_part2 opposite>
C)<body_part1 opposite> <body_part2 correct>
D)<body_part1 opposite> <body_part2 opposite>
```

Figure 9: Prompt templates used for creating PS-Fine. Top: fine-grained generation prompt used to extract body-part–level motion descriptions and their opposites from PoseScript descriptions using LLaMA-3.3-70B. Bottom: judging prompt used to automatically filter incorrect or inconsistent descriptions through multiple-choice verification.

PF-fine. To construct PF-fine, we follow a procedure analogous to PS-fine but applied to the PoseFix [14] dataset, which provides motion descriptions that express how to transition from pose A to pose B. Our goal is to convert these global transition descriptions into fine-grained body-part–level motion primitives that teach the LLM localized motion understanding.

For each transition description, we prompt LLaMA-3.3-70B-Instruct to: (i) select two body parts explicitly mentioned or implied in the sentence, (ii) generate a one-sentence description of how each selected body part moves from pose A to pose B, and (iii) generate an 'opposite' motion description by minimally flipping the movement direction (e.g., forward→backward, lifted→lowered). The model returns two JSON objects, each containing a body part, its fine-grained motion description, and its opposite. These pairs form the PF-fine motion primitives.

As with PS-fine, we apply automatic verification by constructing a multiple-choice judging prompt. Each question contains four options formed by systematically combining correct and 'opposite' motion descriptions across the two body parts. Given the original PoseFix transition description, the LLM must select the correct combined option. Only motion primitives that pass this consistency check are retained. Fig. 10 shows the generation and judging prompts used for constructing PF-fine.

Generation Prompt

```
You are given a description of how to move from pose A to pose B:
<desc_1>

Task:
      • Pick two body parts that are described in the transition.
      • Describe the movement for each chosen body part; always include the
        body part in the sentence.
      • Describe the opposite of the movement by altering only the specific
        difference (e.g., backward instead of forward).

Respond only with two JSONs.  Do not include explanations, reasoning, or
repeat the prompt.
Output format:
{"body_part":  "<chosen body part>",
"description":  "<one short sentence>",
"opposite":  "<one short sentence>"}
{"body_part":  "<chosen body part>",
"description":  "<one short sentence>",
"opposite":  "<one short sentence>"}
```

Judging Prompt

```
You are given a description of moving from pose A to pose B:
<desc_2>

Below are four options describing the movements of <body_part1> and
<body_part2>.
Pick the correct option.  Only return a JSON dictionary of the form:
{"choice":  "a"}

A)<body_part1 correct> <body_part2 correct>
B)<body_part1 correct> <body_part2 opposite>
C)<body_part1 opposite> <body_part2 correct>
D)<body_part1 opposite> <body_part2 opposite>
```

Figure 10: Prompt templates used for creating PF-Fine. Top: generation prompt used to extract body-part–level motion descriptions and their opposites from PoseFix transition descriptions using LLaMA-3.3-70B. Bottom: judging prompt used to verify motion correctness through multiple-choice consistency checking.

B.1.2 PoseScript Question & Answer

We further introduce PS-QA, a machine-generated dataset that produces structured question–answer pairs directly from 3D pose geometry. Unlike PS-Short and PS-Fine, PS-QA does not rely on LLM re-annotation: every answer is computed analytically from joint coordinates. The goal of PS-QA

is to introduce the ability to reason about spatial relations, and identify geometric patterns such as symmetry or joint configuration. PS-QA is built using six complementary question families, each targeting a different aspect of spatial reasoning:

- **Closest Joint.** Given a starting joint (e.g., wrist or ankle), the model must identify which other joint is spatially closest while excluding the joint's direct parent in the kinematic chain. We compute Euclidean distances in 3D and ensure that closest candidates are well-separated to avoid ambiguity.
- **Furthest Joint.** Similar to the above but requiring the identification of the most distant joint relative to a starting joint. Joint distributions are balanced using frequency constraints to avoid bias toward particular anatomical regions.
- **Feet Relation.** This category determines whether the left or right foot is in front or behind. We estimate body orientation using the hip–spine plane and compare the projected foot positions along the forward axis. Only poses with a sufficiently clear front/back difference are included.
- **Symmetry Relation.** We evaluate symmetry by mirroring the left-side joints across the inferred sagittal plane and computing the left–right joint deviations. Depending on the magnitude of these deviations, we label the pose as having *full*, *upper-body*, *lower-body*, or *no* symmetry. This produces categorical answers that test higher-level structural understanding.
- **Joint Bending.** For joints such as the knees, elbows, hips, and spine, we compute bending angles using the standard angle between limb vectors. Angles are discretized into interpretable linguistic categories following [13] (*straight*, *slightly bent*, *partially bent*, *right angle*, *almost completely bent*, *completely bent*). Each category is evenly represented.
- **Comparative Limb Bending.** The model compares the bending of two joints (e.g., left vs. right knee) and decides which is more (or less) bent. Only pairs with a sufficiently large angle difference are retained to avoid uncertain cases.

All questions are phrased using multiple natural-language templates, and the ground-truth answer strings are diversified using paraphrases, while remaining strictly determined by geometry. Each instance additionally includes a small set of multiple-choice alternatives, enabling training and evaluation of both free-form and forced-choice reasoning.

Table 11 summarizes the PS-QA question types and their geometric computations.

Category	**Underlying Computation**
Closest Joint	3D Euclidean distance to a starting joint, parent excluded
Furthest Joint	Maximum 3D distance from starting joint
Feet Relation	Front/back ordering via hip–spine plane projection
Symmetry Relation	Left–right mirrored joint MPJPE thresholds
Joint Bending	Limb angle calculation and discretization
Compare Limb Bending	Relative angle difference between two joints

Table 11: Overview of PS-QA question categories and their geometric computation rules.

B.2 Motion Curriculum

We further detail, BABEL-cap, FTM-QA, and HML3D-QA. For the former two, we highlight that BABEL-QA uses a train/val/test split that does not align with the original BABEL split. To prevent leakage into downstream evaluation, we remove from our BABEL-cap and FTM-QA train/val sets all motion IDs appearing in the BABEL-QA validation or test splits. This guarantees that no evaluation motion is seen during training.

B.2.1 BABEL-cap

To obtain simple motion captions for the early stage of our curriculum, we construct BABEL-cap, a set of short AMASS [48] clips paired with concise BABEL [54] action labels. Each clip is linked to a brief question–answer pair (e.g., *"What action is being performed?"*).

We extract raw action labels from BABEL's `frame_ann` entries when available, otherwise falling back to `seq_ann`. We discard labels that do not describe a meaningful atomic action for short clips, including:

- non-informative labels (`transition`, `unknown`);
- context-dependent labels (e.g., "walk back to", "back to original position");
- segments shorter than two frames.

For each valid action segment, we form a short prompt requesting a minimal action description and pair it with the corresponding BABEL label. We use several interchangeable prompt phrasings (e.g., "Describe the action in a few words."), but do not rely on any complex template logic. Each sample specifies the referenced AMASS motion file, the prompt, the answer, and the annotated time span.

This results in a compact set of atomic action descriptions that serve as the first stage of our motion curriculum.

B.2.2 FTM-QA

FTM-QA extends the temporal supervision available in BABEL by transforming its sequence- and frame-level annotations into structured question–answer pairs. For each motion, we extract the global sequence label together with all frame annotations that include explicit start and end times, ordering them chronologically and discarding non-semantic labels such as `transition` and `unknown`. The remaining labels are formatted into a single annotation block that lists the overall activity followed by each temporally grounded segment. This block forms the sole evidence the model may rely on.

To construct QA pairs, we follow the re-annotation strategy of [10] while adapting it to the richer temporal structure of BABEL. The LLM, Qwen2.5-72B-Instruct, is instructed to generate only questions whose answers can be unambiguously inferred from the motion sequence, without referencing unseen descriptions, relying on the numbering of annotations, or assuming information not encoded in the temporal labels. The prompt highlights typical forms of temporal reasoning—identifying start or end times, determining what action follows or precedes another, relating short events to the global activity, and describing changes over time—while avoiding ambiguity or hallucination. The model returns a JSON list containing the generated question–answer pairs, which we then directly associate with the corresponding AMASS [48] motion clip. This produces a dataset explicitly aligned with temporal ordering, duration reasoning, and segment-level structure, complementing the spatial supervision provided by PS-QA.

Fig. 11 shows the prompt template used for re-annotation.

B.2.3 HML3D-QA

Following prior work [10, 20, 38], we additionally construct question–answer pairs for HumanML3D using an LLM-based procedure. Each HumanML3D sample provides several textual descriptions of the same motion. We follow the established pipeline by supplying these multi-description annotations to an LLM, we use Qwen2.5–72B-Instruct, and prompting it to generate diverse motion QA pairs grounded in the motion itself. The LLM is instructed to: (i) avoid referencing the description text explicitly, (ii) ensure all questions are answerable solely by observing the motion sequence, and (iii) avoid ambiguous or overly speculative questions. The LLM returns a JSON list of QA pairs, which we store directly as the HumanML3D-QA dataset.

Fig. 12 shows the prompt template used for re-annotation.

C Limitations and Broader Impact

Limitations. Although FiGMo supports both 2D and 3D skeletal input, its performance depends on the quality of the input poses. Noisy detections, missing joints, severe occlusion, or unusual

```
This is the complete annotation of one motion sequence.  Sequence label contains
information about the whole sequence, while frame labels contain information about
specific segments indicated by corresponding start-time and end-time.  Sequence labels
can give context to frame labels.  Please construct several QA pairs based on this
information.
!  Note that the sequence label does not have a start or end time.
!  Note that each frame label has its own start and end time, one label can have to do
with a previous one (i.e.  'move back to the original position').
!  Note that don't ask questions about the frame annotations:  'transition' and
'unknown'.
!  Note that you can only see the movement, not the descriptions in advance.  Therefore,
you can NOT ask or answer something like 'the first/third description'.
!  Note that you should only propose questions for which you are sure they are possible
to answer using the motion sequence.  Avoid ambiguity.
!  Note that Don't say anything like 'sure' or 'here is xxx', just return the QA's
directly in the form of a JSON.
Don't rigidly imitate the template either.


HERE IS AN EXAMPLE:
[GIVEN DESCRIPTION]:
Sequence label:
throwing a baseball
Frame labels:
Stand #0.0-0.4
Transition #0.4-0.82
Throw ball with left hand #0.8-2.1
Transition #2.1-2.8
Retreat right foot #2.8-3.7
Stand #3.7-5.0
Walk to left #5.0-7.0


[System output]:
[ {"q":  "What is the main action performed in this sequence?", "a":  "The person throws
a baseball."},
{"q":  "What does the person do after throwing the ball?", "a":  "They retreat their right
foot and return to a standing position."},
{"q":  "At what point does the throwing action begin?", "a":  "Around 0.8 seconds."},
{"q":  "What happens before walking away?", "a":  "The person stands for a moment after
retreating their right foot."},
{"q":  "Which hand is used for the throw?", "a":  "The left hand."},
{"q":  "Does the person walk away immediately after throwing?", "a":  "No, the person
stands for a good second before walking away."},
{"q":  "Does the person walk away, if so in what direction?", "a":  "Yes, the person walks
away to the left."}
]


ACTUAL SAMPLE:
<sample_annotations>
```

Figure 11: Prompt template used to generate FTM-QA. The LLM receives sequence-level and frame-level BABEL annotations and produces temporally grounded question–answer pairs in JSON format.

camera viewpoints may reduce performance. Long sequences are handled through uniform temporal subsampling, which may miss short actions; adaptive temporal sampling is a promising direction for future work. Finally, while we evaluate temporal localization on CompMo and motion QA on multiple benchmarks, broader real-world deployment would require testing across more diverse environments, activities, camera settings, and populations.

Broader impact. Skeletal motion representations can support privacy-preserving applications such as rehabilitation monitoring, sports coaching, and human-robot interaction by reducing reliance on raw video. However, motion-understanding systems could also be misused for surveillance or behavioral monitoring if deployed without consent or appropriate safeguards. Since pose estimators and motion datasets may have demographic or viewpoint biases, downstream systems should be evaluated carefully before deployment in sensitive settings.

```
These are multiple descriptions of the same motion sequence, each line contains a
different description of the motion.  The last two numbers in each line correspond to the
start-time and end-time of the description.  If the numbers are all 0.0, it represents
the entire sequence.  Please construct several motion QA pairs based on this information,
the question should be answerable by only viewing the motion.
!  Note that the start and end time of each line only correspond to the description of
that line, and has nothing to do with the other lines.  Avoid ambiguity.
!  Note that only the movement is provided with the question, NOT the descriptions.
Therefore, you can NOT ask or answer something like 'the first/third description' or
'in the description'.
!  Note that you should not mention anything about the descriptions in the QA's.  Instead
of saying:  'it is described as big' say 'it is big'.
!  Note that you should only propose questions for which you are sure they are possible
to answer using the motion sequence.  Avoid ambiguity.
!  Note that Don't say anything like 'sure' or 'here is xxx', just return the QA's
directly in the form of a JSON.
Don't rigidly imitate the template either.


HERE IS AN EXAMPLE:
[GIVEN DESCRIPTION]:
a person walks toward the front, turns to the right, bounces into a squat, and places
both arms in front of chest before placing them on the knees.#0.0#5.0
person walks up and squats slightly to pose a position#0.0#0.0
he moved forward then stretched his body, moving his hand and touching his knees with
hands and now he turned to the left side.#0.0#0.0
a person walks, turns slightly to the right, squats, puts hand on both knees while
squatting, and then squats again.#0.0#8.0


[System output]:
[ {"q":  "What is the motivation behind the person performing these motions?", "a":
"Without additional context, it is not possible to determine the exact motivation."},
{"q":  "True or False:  The person touches their knees with their hands immediately after
walking.", "a":  "False.  He stretches his body before touching his knees."},
{"q":  "What does the man do after walking forward?", "a":  "The guy turns to the
right."},
{"q":  "What does the guy do after walking forward and turning right?", "a":  "He bounces
into a squat."},
{"q":  "How many times does the man squat?", "a":  "twice."},
{"q":  "Can you describe the initial movement of the person in this sequence?", "a":
"Initially, the person begins by walking forward."},
{"q":  "Could you elaborate on the arm movements of the person during the squat?", "a":
"They place both arms in front of the chest and then on the knees."} ]


ACTUAL SAMPLE:
<sample_annotations>
```

Figure 12: Prompt template used to generate HML3D-QA. The LLM is given multiple natural-language descriptions of the same motion and asked to produce a JSON list of motion-grounded QA pairs.

D Compute Resources

All experiments were run on 4 NVIDIA A100 GPUs with 80 GB memory per GPU. The main FiGMo training required approximately 150 GPU-hours, and each downstream fine-tuning run required approximately 4 GPU-hours. The total compute used for the reported experiments was approximately 478 GPU-hours. Additional preliminary experiments were conducted during development but are not included in this estimate.

E Assets, Licenses, and Release

We use existing datasets and models including AMASS, Human3.6M, PoseTrack, InstaVariety, PoseScript, PoseFix, BABEL, HumanML3D, HuMMan-QA, CompMo, NTU-RGB+D 120, QEVD-Coach, VideoLLaMA3, Qwen2.5, and MMPose. We cite the original sources throughout the paper and use them according to their respective licenses and terms of use. Our derived supervision modules are constructed from these existing resources and will be released together with code and documentation, subject to the licenses and redistribution terms of the underlying datasets. The

released documentation will include data construction procedures, prompt templates, filtering details, and training/evaluation instructions.