

# IMoRe: Implicit Program-Guided Reasoning for Human Motion Q&A

Chen Li<sup>1\*</sup>   Chinthani Sugandhika<sup>1,2\*</sup>   Yeo Keat Ee<sup>1</sup>   Eric Peh<sup>1</sup>   Hao Zhang<sup>1</sup>  
Hong Yang<sup>1</sup>   Deepu Rajan<sup>2</sup>  
Basura Fernando<sup>1,2</sup>

<sup>1</sup>Institute of High-Performance Computing, Agency for Science, Technology and Research, Singapore

<sup>1</sup>Centre for Frontier AI Research, Agency for Science, Technology and Research, Singapore

<sup>2</sup>College of Computing and Data Science, Nanyang Technological University, Singapore

## Abstract

Existing human motion Q&A methods rely on explicit program execution, where the requirement for manually defined functional modules may limit the scalability and adaptability. To overcome this, we propose an implicit program-guided motion reasoning (IMoRe) framework that unifies reasoning across multiple query types without manually designed modules. Unlike existing implicit reasoning approaches that infer reasoning operations from question words, our model directly conditions on structured program functions, ensuring a more precise execution of reasoning steps. Additionally, we introduce a program-guided reading mechanism, which dynamically selects multi-level motion representations from a pretrained motion Vision Transformer (ViT), capturing both high-level semantics and fine-grained motion cues. The reasoning module iteratively refines memory representations, leveraging structured program functions to extract relevant information for different query types. Our model achieves state-of-the-art performance on Babel-QA and generalizes to a newly constructed motion Q&A dataset based on HuMMan, demonstrating its adaptability across different motion reasoning datasets. Code and dataset are available at: <https://github.com/LUNAProject22/IMoRe>.

## 1. Introduction

Understanding human motion has been a crucial challenge in computer vision, with wide applications in human-computer interaction and embodied AI. Significant progress has been made in human motion understanding, especially in action recognition and motion forecasting, while fine-grained human motion understanding and reasoning over subtle motions have been overlooked. Identifying body parts or the direction of movement involved in an action remains challenging due to the lack of high-quality annotated datasets,

\*Equal contribution.

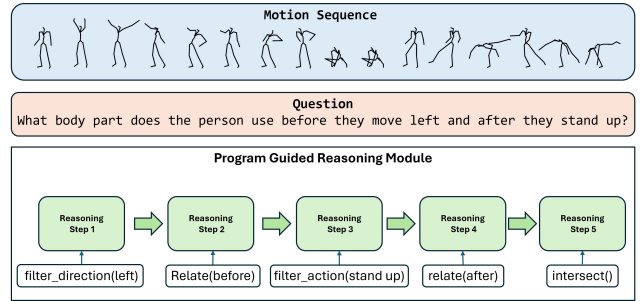


Figure 1. Illustration of the Program-Guided Reasoning Process for Human Motion Q&A. Given a motion sequence and a natural language question, our model executes a structured reasoning process using a program-guided reasoning module.

occlusions, motion ambiguities, variability in human poses, and the difficulty of modeling spatiotemporal dependencies at a fine-grained level. Recently, the human motion Q&A task has been introduced in [11] to address complex and fine-grained human motion understanding. Given a motion sequence and a corresponding question in natural language, this task requires multistep reasoning to predict various attributes of the motion sequence.

Existing approaches [11] to human motion question-answering (Q&A) rely on predefined functional modules that explicitly perform reasoning steps such as filtering, relational inference, and querying. The advantage of explicit program-based reasoning compared to previous end-to-end approaches [33] lies in its interpretability, as it provides access to all intermediate execution results, and in its data efficiency, since each functional module is designed with a small set of parameters. Despite the advantages, these explicit program execution-based methods suffer from limitations in scalability and adaptability due to their reliance on manually defined functional modules. To overcome these constraints, we propose a novel implicit program-guided motion reasoning (IMoRe) framework that uses a unified

reasoning module across multiple types of queries. Instead of leveraging separate handcrafted functional modules, our model dynamically adapts its reasoning process based on structured program functions, ensuring flexibility and generalization across diverse queries.

The specific design of our reasoning module is inspired by the Memory, Attention and Composition (MAC) [16], which conducts iterative reasoning by composing control, read, and write units. The control unit identifies a series of operations from the given text, the read unit extracts relevant information from the visual input to execute each operation, and the write unit iteratively integrates the extracted information into the cell’s memory state, generating intermediate reasoning results. Although the MAC module infers reasoning operations through soft attention over question words, this approach can introduce ambiguity in operation generation, potentially affecting the accuracy of visual information retrieval. To solve this, our model is designed to condition directly on program functions, ensuring a more precise representation of reasoning operations. Furthermore, our approach enhances interpretability by explicitly defining operations, making the reasoning process more transparent.

To further enhance reasoning capacity, we introduce a program-guided reading mechanism. The motion understanding and reasoning task necessitates extracting different attributes from motion sequences. Furthermore, queries about action types require high-level motion sequence understanding, whereas queries about body parts require fine-grained information on localized motion. To address these challenges, we extract multi-level features from a pre-trained motion Vision Transformer (ViT) [38] and allow the reasoning module to select the most appropriate feature level based on the program function. This design allows the model to access the most relevant information dynamically for each reasoning step.

We evaluate our approach on the Babel-QA benchmark, where our model significantly outperforms existing methods. To further test its generalization capability, we introduce a new motion Q&A dataset based on the HuMMan dataset. Additionally, we assess the robustness of our model by using predicted programs, and simulating scenarios where ground truth programs are unavailable. The experimental results demonstrate the effectiveness of our proposed approach across different datasets, highlighting its adaptability and robustness in motion Q&A tasks. Our contributions can be summarized as follows:

- We propose a novel implicit program-guided reasoning framework that unifies reasoning across multiple operation types without relying on predefined functional modules.
- We introduce a program-guided reading mechanism that enables the model to dynamically choose the most suitable features for each step of reasoning.
- We achieve state-of-the-art performance on the Babel-QA

benchmark and introduce a new motion Q&A dataset based on HuMMan to evaluate generalization capability.

## 2. Related Work

**Human Motion Understanding** Human motion understanding has been widely studied, particularly in action recognition [2–4, 7, 8, 23]. ST-GCN [34] introduced graph convolutional networks (GCNs) to model spatial-temporal relationships between body joints, establishing a dominant paradigm in skeleton-based action recognition. Subsequent works have improved GCN architectures to enhance accuracy. Beyond GCNs, PoseConv3D [10] proposed a 3D heatmap volume representation, enabling 3D-CNN models to achieve superior performance. For untrimmed motion sequences, temporal convolutional networks (TCN) [12, 35] and transformer-based models [32] estimate per-frame action probabilities to facilitate action localization. Further, motion-language research has gained attention in linking natural language with 3D human motion. Notable efforts include various motion-language datasets such as KIT [30], HumanML3D [13], HuMMan [39]. Existing motion-language models focus mainly on contrastive learning to align motion and text representations [33, 38]. While these models provide valuable insights into motion sequences, they lack the ability to perform multi-step reasoning that combines action-level understanding with fine-grained motion analysis, such as identifying body parts involved in specific frames or detecting sudden changes in movement. To address this limitation, the recent work NSPose [11] introduced the human motion question-answering (QA) task, which evaluates fine-grained human behavior understanding by pairing motion sequences with natural language questions and categorical answers. Building on this foundation, our study also addresses the human motion QA problem, aiming to enhance human motion reasoning by developing a more adaptive and scalable reasoning framework.

**Reasoning Approaches for Question Answering** Question Answering (QA) in Artificial Intelligence is a fundamental research area focused on developing systems capable of understanding and responding to user queries across various modalities, including text, images, video, motion, and multimodal inputs. Early QA systems were mainly built on rule-based or statistical methods [20]. However, with the advent of deep learning, neural reasoning has become a dominant approach. Recent advances in neural reasoning for QA span multiple modalities, including text-based QA [28, 36], image-text QA [21, 22, 25, 26], and video-text QA [19, 40]. Most of these models are built upon Recurrent Neural Networks (RNNs) and Transformer architectures. Meanwhile, neuro-symbolic reasoning has demonstrated significant success in visual reasoning tasks [17, 27, 37].

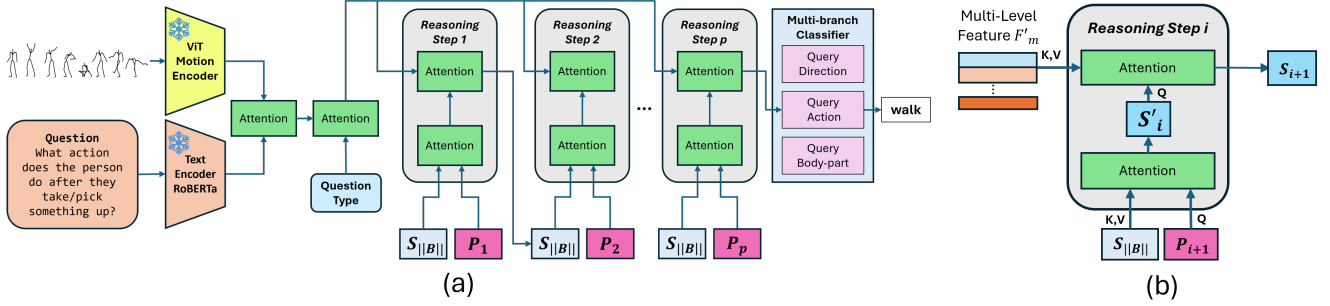


Figure 2. (a) Overall pipeline of our Program-Guided Motion Reasoning Framework. Motion and text features are extracted using a ViT-based motion encoder and RoBERTa. These features are processed through a structured reasoning module, where intermediate memory states are iteratively refined using program guidance. A multi-branch classifier is adopted to predict the final answer. (b) Illustration of one reasoning step.

Neuro-symbolic VQA [37] integrates symbolic program execution with visual recognition to improve question answering performance, achieving strong performance on the CLEVR benchmark [18]. NS-CL [15] further improved this paradigm by eliminating the need for dense supervision, enabling the model to learn reasoning patterns by observing images and reading natural language questions and answers. MMN [5] improves over Neural Module Network [1] by taking in function recipes and morph into diverse instance modules dynamically. More recently, neuro-symbolic frameworks have been extended to temporal reasoning [6], knowledge-based reasoning [29] and 3D reasoning [14, 15] as well. In human-motion reasoning, NSPose employs a neuro-symbolic reasoning model that explicitly defines functional modules to process motion data. While this approach offers advantages such as interpretability and data efficiency, its reliance on manually defined functional modules poses challenges in terms of scalability and adaptability, particularly when handling diverse and complex motion scenarios. Overcoming the above limitations, we propose a neural-based implicit program-guided reasoning approach for human motion Q&A. Unlike explicit neuro-symbolic methods, our approach learns to reason implicitly without predefined functional operations, allowing for greater flexibility, scalability, and adaptability in complex motion reasoning tasks.

### 3. Method

#### 3.1. Overview

We propose a novel memory mechanism-based reasoning framework for the human motion Q&A task. This framework is designed to utilize structured program functions to guide the reasoning process, eliminating the need for manually defined functional modules. This approach enables interpretable and flexible reasoning over human motion sequences. The overall framework is illustrated in Fig. 2 (a). Given a human motion sequence represented by 3D joint

locations as  $\mathcal{M} \in \mathcal{R}^{T \times J \times 3}$ , where  $T$  is the number of frames and  $J$  denotes the number of joints per frame, and a natural language question  $Q$ , the objective is to predict an accurate answer by reasoning over the provided motion sequence. The motion sequence is first processed by a motion encoder, which extracts feature embeddings representing spatial-temporal dynamics. Additionally, we assume the availability of a structured program  $\mathcal{P}$ , which can either be a ground-truth program or a predicted program from an existing program predictor [9]. Rather than executing directly over the feature representation in a rule-based manner as done in [11], we have developed a unified reasoning module inspired by the MAC architecture [16]. The original MAC model generates operations by computing soft attention over question words, which introduces potential ambiguity due to language variability and contextual dependencies. To mitigate these limitations, we introduce a program-guided reasoning module. The program guidance ensures that each reasoning step follows an explicitly defined program function, hence eliminating inconsistencies introduced by soft-attention-based reasoning. This reasoning process is further elaborated in Section 3.3. Additionally, different types of questions require reasoning over different motion attributes. For instance, queries about action types demand a high-level semantic understanding of motion sequences, while queries about body parts necessitate fine-grained spatial attention to localized movements. To effectively connect these hierarchical representations, we introduce a program-guided reading mechanism that dynamically selects the most informative motion features for each reasoning step.

#### 3.2. Preliminaries on MAC Network

The Memory-Attention-Control (MAC) network is a neural reasoning architecture designed for compositional reasoning in question-answering tasks. The MAC network operates iteratively, decomposing complex reasoning into multiple

stepwise operations, where each step refines an intermediate memory state to arrive at the final answer. A MAC network consists of three key components: 1) Control Unit which determines the reasoning operation at each step by attending to relevant parts of the question using a soft attention mechanism; 2) Read Unit which extracts pertinent information from the visual input based on the control signal, allowing iterative refinement of retrieved evidence; 3) Write Unit which integrates the extracted information into a recurrent memory state, progressively enhancing the representation of the answer.

The iterative reasoning of the MAC network enables multi-step inference, making it particularly suitable for tasks requiring compositional logic, such as visual question answering (VQA). However, despite its effectiveness in structured reasoning, MAC generates operations based on soft attention over question words, which introduces ambiguity and variability in operation at each reasoning step.

### 3.3. Implicit Program-Guided Reasoning Module

Given the motion sequence  $\mathcal{M}$  and question  $\mathcal{Q}$ , we first use a motion encoder  $\mathbf{E}_m$  and a text encoder  $\mathbf{E}_t$  to extract feature embeddings for the motion sequence and question respectively. We adopted the pre-trained ViT model from [38] as the motion encoder, which introduces a patch-based representation for the skeleton structure. Instead of considering each joint, the patch-based representation divides the skeleton joints into body parts, hence better encoding skeleton structure information. The RoBERTa representation in [24] is used as our text encoder. The motion feature  $f_m = \mathbf{E}_m(\mathcal{M})$  and the text feature  $f_t = \mathbf{E}_t(\mathcal{Q})$  are first fused to obtain a text-aware motion feature:

$$h_m = \text{Attention}(Q = f_m, K = f_t, V = f_t). \quad (1)$$

We use cross-attention with the query being the motion feature and the key and value being the text feature. In addition to fusing with the text, the motion feature is further fused with the question type information using the same attention mechanism, with query being  $h_m$  and key and value being the question type. The question type consists of query direction, query action and query body part, and is encoded with the RoBERTa before fusion. The final fused motion feature is used for the iterative reasoning process.

We use the program as the guidance for reasoning. The program is represented as  $\mathcal{P} \in \mathcal{R}^{p \times d}$ , where  $p$  denotes the length of the program and  $d$  the program embedding dimension. Note that the program can be ground truth as used in [11] or predicted from an existing program predictor [9]. At each reasoning step, we gradually refine the intermediate memory states guided by the program. Specifically, at the  $(i + 1)$ -th reasoning step with previous memory states being  $\{S_1, S_2 \dots S_i\}$ , we first attend to the previous states that are related to the current reasoning step by an attention

mechanism:

$$S'_i = \text{Attention}(Q = P_{i+1}, K = S_{\parallel \mathcal{B} \parallel}, V = S_{\parallel \mathcal{B} \parallel}), \quad (2)$$

where  $P_{i+1}$  represents the program function for the  $(i + 1)$ -th reasoning step and  $S_{\parallel \mathcal{B} \parallel}$  represents the previous memory states that are related to the current step. For example, the question “What action does the person do before they move left?” and the corresponding program “*query\_action(related(before, filter(left)))*”, the program function *related()* depends on the output of *filter()*. It is important to encode this dependency information to provide context information for each reasoning step. The dependency relationship is provided by the structure information of the program.

In addition to the previous context information, the model also requires information to be extracted from the input motion sequence to conduct reasoning. This process is similar to the reading stage of the MAC, where the key is to extract the most relevant information for the current reasoning step. We achieve this by directly taking the guidance from program functions. For example, for the question “What action does the person do before they move left?”, the program functions *filter(move left)* or *related(before)* provides clear information on which part to attend. This process is implemented as an attention module with  $S'_i$  being the query and motion feature  $h_m$  being the key and query:

$$S_{i+1} = \text{Attention}(Q = S'_i, K = h_m, V = h_m). \quad (3)$$

The intuition is to refine the current memory states  $S'_i$  with relevant motion features in  $h_m$ . Compared to operation generated by soft attention over question words in the control unit of MAC, the direct program information provides a more precise representation. As a consequence, the proposed program-guided reasoning mitigates operation ambiguities by directly conditioning each reasoning step on explicit program functions.

### 3.4. Program-Guided Reading Mechanism

We introduce a program-guided reading mechanism to enhance the reasoning capacity of our model by enabling dynamic retrieval from a knowledge pool. Unlike the MAC model, which extracts information from a single static feature representation, our model constructs a multi-level knowledge pool by aggregating intermediate representations from different stages of the Vision Transformer (ViT). This allows the reasoning module to adaptively select the most relevant feature representation at each step based on the program function. We define the knowledge pool as a collection of hierarchical feature embeddings extracted from layers of the ViT model  $F_m = \{h_i\}_{i=1}^M$ , where  $h_i$  represents the feature embeddings from the  $i$ -th intermediate layer of ViT. Since feature representations from different layers encode



varying levels of abstraction, lower-layer features capture fine-grained local motion details, while higher-layer features encode global semantic information. Different levels of features are important for different motion concepts. For example, the question ‘‘What body part does the person use after they jump’’ requires both high-level semantic understanding for the concept ‘jump’ and low-level local information for moving ‘body part’. We apply a projection layer to ensure all feature embeddings reside in a shared latent space:

$$h'_i = W_i^p h_i, \quad \forall i \in 1, \dots, M, \quad (4)$$

where  $W_i^p$  is the projection layer for the  $i$ -th intermediate output. Our program-guided reading mechanism is implemented by replacing the single motion feature  $h_m$  in Eqn.(3) with the multi-level feature  $F'_m = \{h'_i\}_{i=1}^M$ , expressed as

$$\mathcal{S}_{i+1} = \text{Attention}(Q = \mathcal{S}'_i, K = \{h'_i\}_{i=1}^M, V = \{h'_i\}_{i=1}^M). \quad (5)$$

We also illustrate the detailed process of one reasoning step in Fig. 2 (b). By incorporating the most informative feature representation at each step, our model can effectively capture dependencies across hierarchical motion features, making it highly adaptable to diverse motion concepts.

### 3.5. Iterative Reasoning

The reasoning module executes for  $p$  iterative steps, progressively refining the memory state until reaching the final representation. Each intermediate state is stored in a memory matrix, allowing the model to leverage dependency information between reasoning steps, as formulated in Eqn (2). The iterative refinement process is formulated as:

$$\mathcal{S}_i = f(\mathcal{S}_{i-1}, \mathcal{P}_i, F'_m), \quad i = 1, 2, \dots, p, \quad (6)$$

where  $f$  represents the reasoning function integrating prior memory  $\mathcal{S}_{i-1}$ , the program function  $\mathcal{P}_i$ , and motion features  $F'_m$ . Once the final reasoning step is completed, the resulting representation is passed into a **multi-branch classifier**, where each branch specializes in predicting answers corresponding to a specific question type, including query\_action, query\_direction, and query\_body\_part. For a given question, only the corresponding branch depending on the question type is activated. The question type information can be directly obtained from the provided program.

### 3.6. Training and Optimization

The objective of our model is to predict the correct answer based on the input motion sequence and question. We employ a classification-based answer prediction framework, where the model predicts a probability distribution over the candidate answers in the dataset. The predicted answer vector  $v^{\text{ans}} \in \mathcal{R}^a$  represents the confidence scores for  $a$  possible answers. The loss function of our model can be defined as:

$$\mathcal{L} = -\log p(v^{\text{ans}} | \mathcal{M}, \mathcal{Q}, \mathcal{P}). \quad (7)$$

We use the AdamW optimizer with weight-decay regularization to prevent over-fitting, the weight-decay coefficient is set to 0.0001.

## 4. Experiments

**Training Details** We used the pre-trained motion ViT encoder in [38] to extract the multilevel motion features required for the program-guided reading mechanism of our IMoRe. This converts raw motion sequences into a unified representation known as motion patches, which serve as the input for the ViT-based motion encoder, with each patch encapsulating joints relevant to specific body parts such as the torso, right leg, and left arm. We used the RoBERTa [24] text encoder to extract the required text embeddings. The network was trained for 100 epochs using the AdamW optimizer with a learning rate of 1e-6, dropout of 0.1, and batch size 4 on a 48 GB NVIDIA RTX 6000 GPU. All experiments were repeated three times, and the average results were reported.

**Implementation Details** We implement our approach under two settings as, (1) **IMoRe I**: In this setting, we provide the entire human motion sequence as input. Since the motion ViT encoder supports segments of 224 frames, we partition the sequence accordingly and pass each segment separately through the encoder covering the entire motion sequence. (2) **IMoRe II**: In this setting, rather than processing the complete sequence, we randomly sample 224 frames from the whole sequence with a random starting index following [38]. During validation and testing, we execute five runs for the same sequence and select the run with the highest logits scores. The intuition behind this setting is that the segment with the highest score should include the segment with the query concept. We construct the composite knowledge pool required for the program-guided reading mechanism described in Section 3.4 by extracting features from  $M = 6$  intermediate layers (i.e.  $i \in \{0, 2, 4, 6, 8, 11\}$ ) of motion ViT along with its final encoder output. Consequently, each 224-frame segment is processed through these six intermediate layers and the final encoder to form the composite knowledge pool.

We further evaluate our approach under two scenarios: one with ground truth programs provided and one with predicted programs. For the latter scenario, we adapt the coarse-to-fine two-stage program generation network from [9] to predict the required programs. This experiment is to show the robustness of our method to noisy programs since the ground truth program is not always available in practice. Experimental results denoted with an asterisk mark (\*) representing the performance with predicted programs.

**Datasets.** We evaluate our model on two datasets Babel-QA [11] and HuMMan-QA. (1) **Babel-QA**: consists of 1800 training questions, 384 validation questions, and 393 test questions categorized into action, direction, and body part-related questions extracted from 771 motion sequences. (2) **HuMMan-QA**: Given the small size of the Babel-QA dataset and our goal to test the generalizability of our model across diverse datasets, we developed the HuMMan-QA dataset based on the HuMMan-MoGen dataset [39]. The HuMMan-MoGen dataset comprises 6,264 SMPL motion sequences from 160 actions, each annotated with 112,112 detailed temporal and spatial text descriptions. We utilized the motion QA generation strategy in [11]. However, since HuMMan-MoGen lacks segment-level labels for actions, directions, and body parts, we employed the GPT-4o model with in-context learning examples to generate these labels, which were then meticulously refined and manually verified against the actual motion sequences by the authors. The HuMMan-QA dataset consists of 2066 training questions, 524 validation questions, and 533 test questions, with 158, 18, and 6 classes for action, body parts, and direction, respectively, derived from 1,311 motion sequences. Accuracy is used as the evaluation metric.

#### 4.1. Results on Babel-QA Dataset

We report the results of our IMoRe on the Babel-QA dataset in Table 1. We compare our approach with state-of-the-art approaches, including NSPose [11], which is a neural symbolic approach with pre-define functional modules, and CLIP, 2S-AGCN-MLP, 2S-AGCN-RNN, MotionCLIP-MLP, and MotionCLIP-RNN [11], which directly fuses motion and text features. The results reveal several key insights. Firstly, the neural symbolic-based approaches (NSPose, IMoRe) have better performance compared with direct feature fusion based approaches (2S-AGCN-MLP, 2S-AGCN-RNN, MotionCLIP-MLP, MotionCLIP-RNN). This shows the effectiveness of the neural symbolic based approach for the motion reasoning task. Secondly, it can be seen that IMoRe I and IMoRe II outperform the state-of-the-art NSPose [11] by a significant margin for the ground truth and predicted program setting. This demonstrates the effectiveness of our implicit program-guided reasoning approach for human motion reasoning. Lastly, while our IMoRe experiences only a modest performance drop (1-2%) in the predicted program setting, NSPose exhibits a substantial decline in performance. We attribute this to the fundamental differences between explicit and implicit reasoning approaches. In explicit reasoning, functional programs are highly dependent on the correct execution of each step: an error in any predicted step can cascade into a significant performance drop. In contrast, implicit reasoning provides greater adaptability, allowing the model to recover from incorrect input signals and mitigate performance degradation. This resilience highlights

the robustness of our implicit approach in handling noisy or imperfect program predictions, making it more suitable for real-world scenarios where execution errors are inevitable.

#### 4.2. Generalization to HuMMan-QA Dataset

We further evaluate the generalizability of our approach on the HuMMan-QA dataset, with results presented in Table 2. We can see that both IMoRe I and IMoRe II consistently outperform existing models, including both neural symbolic-based approach NSPose and direct motion-text feature fusion-based approach MotionCLIP-MLP and MotionCLIP-RNN, achieving significant performance improvements of 4% and 5.6%, respectively. This demonstrates the ability of our network to generalize effectively across different datasets. Additionally, we observe that while MotionCLIP-MLP does not perform as well on the Babel-QA dataset, it achieves competitive results on HuMMan-QA. This suggests that MotionCLIP-MLP might be particularly suited for the characteristics of the HuMMan-QA dataset, highlighting the importance of dataset-specific model adaptability. These results highlight the effectiveness of our program-guided implicit reasoning approach and its capacity to generalize to different motion question-answering benchmarks. It should be noted that performance across all models in ‘Direction: BTW’ type questions is zero because no question is generated for this type for the HuMMan-QA Dataset.

#### 4.3. Ablation Study

We perform an ablation study on the Babel-QA dataset to verify the effectiveness of each component. We evaluate four models: A. ViT + explicit reasoning (NSPose); B. ViT + implicit reasoning (MAC); C. Implicit program-guided reasoning; D. Implicit Program-guided reasoning + feature selection (i.e., our full model). The first model replaces the motion encoder used in NSPose with ViT to show that our performance gain is not coming from a different motion encoder. The second model is to show the effectiveness of implicit reasoning. We re-implemented the MAC [16] network on the Babel-QA dataset. The third model demonstrates the benefit of program guidance, while the fourth highlights the advantage of program-guided feature selection.

The results are shown in Table 3, from which we can draw several conclusions. First, the performance of NSPose drops when replacing the encoder with the ViT encoder, which shows that the better performance of our IMoRe comes from a better reasoning module. Second, implicit reasoning yields improved results compared to explicit reasoning. Despite the simplicity of MAC’s design, its implicit reasoning mechanism surpasses NSPose’s explicit reasoning. This advantage likely stems from the inherent adaptability of implicit reasoning, which allows the model to handle diverse reasoning tasks more effectively without relying explicitly on prede-

Model	Overall	Query action				Query direction				Query body part			
		All	Before	After	BTW	All	Before	After	BTW	All	Before	After	BTW
CLIP [31]	0.417	0.467	0.380	0.452	0.591	0.366	0.467	0.292	0.222	0.261	0.261	0.278	<u>0.333</u>
2S-AGCN-MLP [11]	0.355	0.384	0.353	0.411	0.273	0.352	0.378	0.250	0.278	0.228	0.261	0.130	<u>0.333</u>
2S-AGCN-RNN [11]	0.357	0.396	0.349	0.396	0.409	0.352	0.400	0.396	0.278	0.194	0.261	0.111	0.167
MOTIONCLIP-MLP [11]	0.430	0.485	0.411	0.470	0.545	0.361	0.400	0.271	0.333	0.272	0.304	0.222	<u>0.333</u>
MOTIONCLIP-RNN [11]	0.420	0.489	0.461	0.441	0.606	0.310	0.400	0.333	0.222	0.250	0.333	0.167	<u>0.333</u>
NSPOSE [11]	0.578	0.627	0.618	0.620	0.639	0.598	0.389	<b>0.583</b>	<u>0.750</u>	0.325	0.296	<b>0.471</b>	0.083
NSPOSE* [11]	0.472	0.519	0.570	0.493	0.500	0.488	0.375	0.238	0.375	0.189	0.250	0.176	0.167
IMoRe I	0.609	0.652	<u>0.640</u>	0.676	<u>0.722</u>	0.622	0.486	0.393	0.458	<u>0.373</u>	<u>0.389</u>	0.353	<b>0.440</b>
IMoRe I*	0.602	0.646	<u>0.640</u>	0.671	0.713	0.611	<u>0.514</u>	0.357	0.375	0.351	0.380	0.343	0.250
IMoRe II	<b>0.640</b>	<b>0.695</b>	<b>0.677</b>	<b>0.707</b>	<b>0.750</b>	<b>0.679</b>	0.458	<u>0.560</u>	<b>0.792</b>	0.358	<b>0.407</b>	<u>0.441</u>	0.083
IMoRe II*	<u>0.615</u>	0.656	0.610	<u>0.700</u>	0.667	<u>0.624</u>	<b>0.528</b>	0.429	0.583	<b>0.391</b>	<u>0.389</u>	<u>0.441</u>	<u>0.333</u>

Table 1. Evaluation of IMoRe and existing models on Babel-QA test set. The asterisk mark (\*) refers to the setting where the model uses the predicted programs as the input. BTW refers to ‘Between’. The bold and underline font shows the best and the second best result respectively.

Model	Overall	Query action				Query direction				Query body part			
		All	Before	After	BTW	All	Before	After	BTW	All	Before	After	BTW
MotionCLIP-MLP [11]	0.686	0.682	0.640	0.628	0.467	0.750	<u>0.500</u>	<b>1.000</b>	0.000	<u>0.692</u>	<b>0.696</b>	0.644	0.639
MotionCLIP-RNN [11]	0.623	0.601	0.559	0.541	0.583	0.750	<b>1.000</b>	0.500	0.000	0.667	<u>0.638</u>	0.616	0.694
NSPose [11]	0.691	0.700	<b>0.686</b>	0.610	0.729	<u>0.822</u>	0.425	<u>0.833</u>	0.000	0.677	0.620	0.639	0.833
IMoRe I	<u>0.719</u>	<u>0.744</u>	<u>0.652</u>	<u>0.734</u>	<b>0.854</b>	<b>1.000</b>	<b>1.000</b>	<b>1.000</b>	0.000	0.665	0.609	<u>0.647</u>	<b>0.889</b>
IMoRe II	<b>0.730</b>	<b>0.746</b>	0.648	<b>0.739</b>	<u>0.813</u>	<b>1.000</b>	<b>1.000</b>	<b>1.000</b>	0.000	<b>0.717</b>	0.636	<b>0.703</b>	<u>0.861</u>

Table 2. Evaluating the generalizability of our approach on HuMMan-QA dataset. BTW refers to ‘Between’. There is no question generated for ‘Between’ relation for query direction in this dataset. The bold and underline font shows the best and the second best result respectively.

finer function modules. Third, the better performance of program-guided reasoning compared to MAC verifies the effectiveness of the program guidance. Third, the incorporation of program-guided feature selection leads to notable performance improvements of 5.5%. This highlights the advantage of extracting task-relevant motion features based on the program information. For instance, action-type queries benefit from high-level motion features, while body-part-related queries require fine-grained motion representations. By dynamically selecting relevant features, our implicit reasoning with program-guided feature selection effectively tailors feature extraction to the specific reasoning task.

#### 4.4. Qualitative Results

The qualitative examples in Fig. 3 illustrate the effectiveness of our proposed IMoRe method in understanding motion sequences with relational reasoning across different aspects: action, direction, and body part. Compared to NSPose, our model consistently provides more accurate reasoning results across various relational queries. In the first example (top left), when asked what action occurs before “taking/picking something up”, IMoRe correctly predicts “squat,” whereas NSPose incorrectly outputs “walk” which occurred earlier in the sequence. This highlights IMoRe’s ability to capture fine-grained, preparatory actions that are crucial for understanding human motion. In the example of query direction in between two movements, while NS-Pose wrongly predicts

“left”, IMoRe correctly predicts “right” showcasing its ability to model spatial transitions. In the query body part “before” relation case, IMoRe correctly identifies that the “right hand” is used before the “right foot”, while NS-Pose incorrectly predicts the “right leg”. This indicates that IMoRe effectively captures body part dependencies by dynamically attending to relevant motion frames, whereas NS-Pose fails to focus on the correct segment. Despite its strong performance, IMoRe still encounters challenges in certain cases. One notable failure occurs in queries involving body parts with the “in-between” relation, where the model struggles to differentiate between closely related actions. For example, when asked about the body part used after one action and before another, IMoRe struggles likely due to the confusion of similar concepts. Another failure case appears in movement direction queries with the “after” relation, where both IMoRe and NSPose incorrectly infer the *forward* direction instead of recognizing the subtle directional change present in the motion sequence.

## 5. Conclusion

We presented an implicit program-guided reasoning framework for human motion Q&A, addressing the limitations of explicit program execution. Our method dynamically adapts reasoning based on structured program functions to improve scalability and adaptability. The program-guided reading

Model	Overall	Query action				Query direction				Query body part			
		All	Before	After	BTW	All	Before	After	BTW	All	Before	After	BTW
A. ViT + NSPose [11]	0.440	0.438	0.433	0.452	0.211	0.450	0.300	0.270	0.000	<b>0.362</b>	<b>0.476</b>	0.258	<b>0.667</b>
B. ViT + MAC [16]	0.582	0.649	0.597	0.633	<u>0.712</u>	0.526	0.356	<u>0.458</u>	<u>0.500</u>	0.356	0.406	<u>0.315</u>	<u>0.083</u>
C. Program-guided reasoning	0.607	0.676	0.632	0.697	0.704	0.571	0.403	0.321	<u>0.500</u>	0.340	0.352	<b>0.441</b>	<u>0.083</u>
D. Program-guided reasoning + feature selection	<b>0.640</b>	<b>0.695</b>	<b>0.677</b>	<b>0.707</b>	<b>0.750</b>	<b>0.679</b>	<b>0.458</b>	<b>0.560</b>	<b>0.792</b>	<u>0.358</u>	<u>0.407</u>	<b>0.441</b>	0.000

Table 3. Ablation study on Babel-QA dataset for explicit reasoning, implicit reasoning, implicit reasoning without program-guided feature selection, and implicit reasoning with program-guided feature selection. BTW refers to ‘Between’. The bold and underline font shows the best and the second best result, respectively.

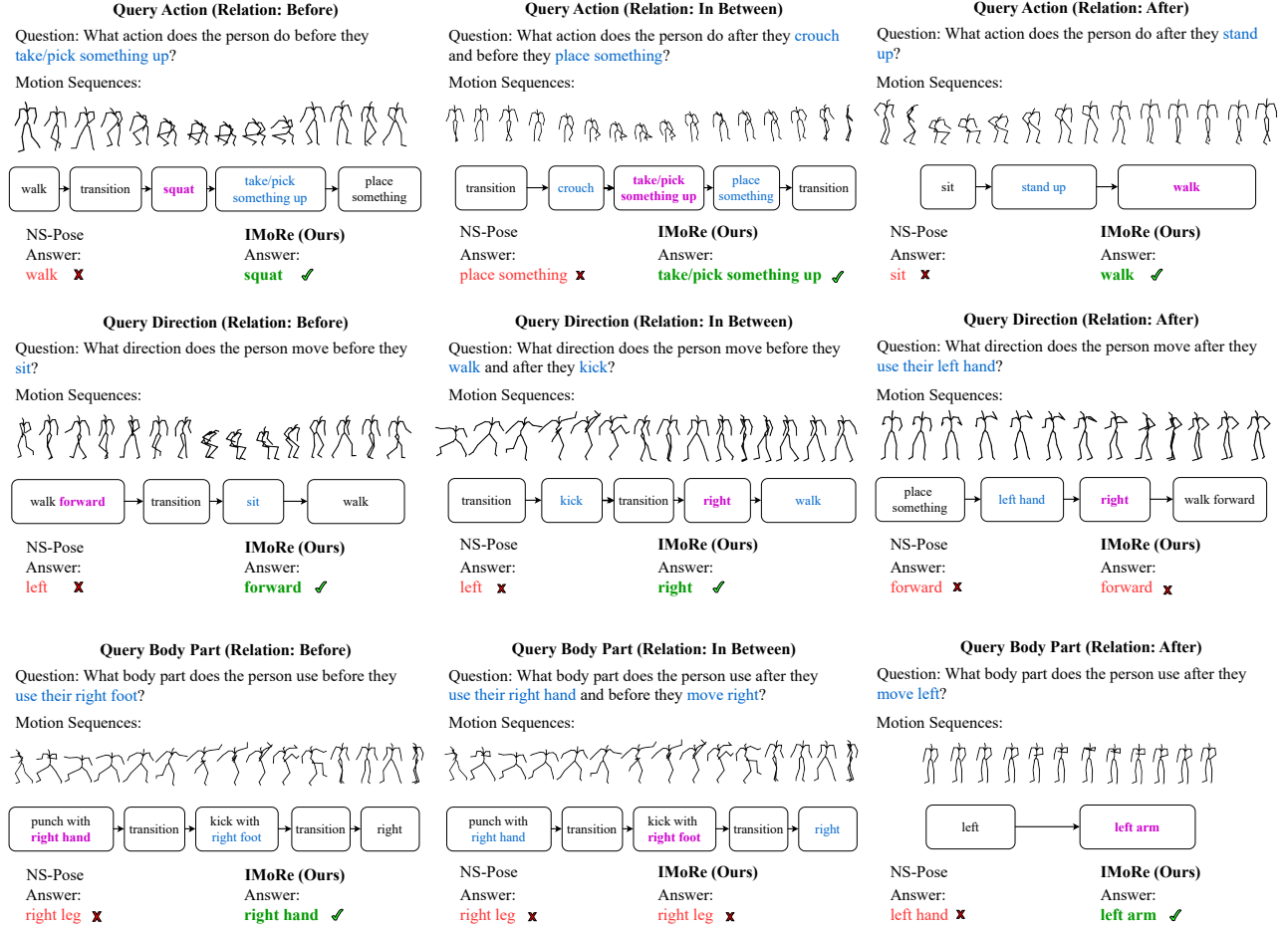


Figure 3. Qualitative comparison between our method (IMoRe) and NSPose on motion sequence reasoning tasks with different query types: action, direction, and body part with different temporal relations including before, in between, and after.

mechanism enables the dynamic selection of multi-level motion features from a pre-trained ViT model, allowing the model to reason over both high-level semantics and fine-grained motion cues. Our model achieves state-of-the-art performance on the Babel-QA dataset and generalizes to the newly constructed HuMMAN-QA dataset. Analysis with predicted programs confirms the robustness and effectiveness of our approach in addressing diverse motion-related queries in real-world scenarios.

**Acknowledgments** This research/project is supported by the National Research Foundation, Singapore, under its NRF Fellowship (Award# NRF-NRFF14-2022-0001) and by funding allocation to Basura Fernando by the A\*STAR under its SERC Central Research Fund (CRF), as well as its Centre for Frontier AI Research.



## References

- [1] Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. Neural module networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 39–48, 2016. 3
- [2] Sadjad Asghari-Esfeden, Mario Sznaier, and Octavia Camps. Dynamic motion representation for human action recognition. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 557–566, 2020. 2
- [3] Carlos Caetano, Jessica Sena, François Brémont, Jefersson A Dos Santos, and William Robson Schwartz. Skelemotion: A new representation of skeleton joint sequences based on motion information for 3d action recognition. In *2019 16th IEEE international conference on advanced video and signal based surveillance (AVSS)*, pages 1–8. IEEE, 2019.
- [4] Jinmiao Cai, Nianjuan Jiang, Xiaoguang Han, Kui Jia, and Jiangbo Lu. Jolo-gcn: mining joint-centered light-weight information for skeleton-based action recognition. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 2735–2744, 2021. 2
- [5] Wenhui Chen, Zhe Gan, Linjie Li, Yu Cheng, William Wang, and Jingjing Liu. Meta module network for compositional visual reasoning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 655–664, 2021. 3
- [6] Zhenfang Chen, Jiayuan Mao, Jiajun Wu, Kwan-Yee Kenneth Wong, Joshua B Tenenbaum, and Chuang Gan. Grounding physical concepts of objects and events through dynamic visual reasoning. *arXiv preprint arXiv:2103.16564*, 2021. 3
- [7] Ke Cheng, Yifan Zhang, Xiangyu He, Weihang Chen, Jian Cheng, and Hanqing Lu. Skeleton-based action recognition with shift graph convolutional network. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 183–192, 2020. 2
- [8] Vasileios Choutas, Philippe Weinzaepfel, Jérôme Revaud, and Cordelia Schmid. Potion: Pose motion representation for action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7024–7033, 2018. 2
- [9] Li Dong and Mirella Lapata. Coarse-to-fine decoding for neural semantic parsing. *arXiv preprint arXiv:1805.04793*, 2018. 3, 4, 5
- [10] Haodong Duan, Yue Zhao, Kai Chen, Dahua Lin, and Bo Dai. Revisiting skeleton-based action recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2969–2978, 2022. 2
- [11] 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. 1, 2, 3, 4, 6, 7, 8
- [12] Benjamin Filtjens, Bart Vanrumste, and Peter Slaets. Skeleton-based action segmentation with multi-stage spatial-temporal graph convolutional neural networks. *IEEE Transactions on Emerging Topics in Computing*, 12(1):202–212, 2022. 2
- [13] Chuan Guo, Shihao Zou, Xinxin Zuo, Sen Wang, Wei Ji, Xingyu Li, and Li Cheng. Generating diverse and natural 3d human motions from text. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5152–5161, 2022. 2
- [14] Yining Hong, Yilun Du, Chunru Lin, Josh Tenenbaum, and Chuang Gan. 3d concept grounding on neural fields. *Advances in Neural Information Processing Systems*, 35:7769–7782, 2022. 3
- [15] Joy Hsu, Jiayuan Mao, and Jiajun Wu. Ns3d: Neuro-symbolic grounding of 3d objects and relations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2614–2623, 2023. 3
- [16] Drew A Hudson and Christopher D Manning. Compositional attention networks for machine reasoning. *arXiv preprint arXiv:1803.03067*, 2018. 2, 3, 6, 8
- [17] Shantanu Jaiswal, Debaditya Roy, Basura Fernando, and Cheston Tan. Learning to reason iteratively and parallelly for complex visual reasoning scenarios. In *Advances in Neural Information Processing Systems*, pages 137965–137998. Curran Associates, Inc., 2024. 2
- [18] Justin Johnson, Bharath Hariharan, Laurens Van Der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2901–2910, 2017. 3
- [19] Seonhoon Kim, Seohyeong Jeong, Eunbyul Kim, Inho Kang, and Nojun Kwak. Self-supervised pre-training and contrastive representation learning for multiple-choice video qa. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 13171–13179, 2021. 2
- [20] Bernardo Leite and Henrique Lopes Cardoso. Do rules still rule? comprehensive evaluation of a rule-based question generation system. In *CSEDU (2)*, pages 27–38, 2023. 2
- [21] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International conference on machine learning*, pages 12888–12900. PMLR, 2022. 2
- [22] Ji Lin, Hongxu Yin, Wei Ping, Pavlo Molchanov, Mohammad Shoeybi, and Song Han. Vila: On pre-training for visual language models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 26689–26699, 2024. 2
- [23] Chunhui Liu, Yueyu Hu, Yanghao Li, Sijie Song, and Jiaying Liu. Pku-mmd: A large scale benchmark for skeleton-based human action understanding. In *Proceedings of the workshop on visual analysis in smart and connected communities*, pages 1–8, 2017. 2
- [24] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019. 4, 5
- [25] Lin Ma, Zhengdong Lu, and Hang Li. Learning to answer questions from image using convolutional neural network. In *Proceedings of the AAAI conference on artificial intelligence*, 2016. 2

- [26] Mateusz Malinowski, Marcus Rohrbach, and Mario Fritz. Ask your neurons: A neural-based approach to answering questions about images. In *Proceedings of the IEEE international conference on computer vision*, pages 1–9, 2015. [2](#)
- [27] Jiayuan Mao, Chuang Gan, Pushmeet Kohli, Joshua B Tenenbaum, and Jiajun Wu. The neuro-symbolic concept learner: Interpreting scenes, words, and sentences from natural supervision. *arXiv preprint arXiv:1904.12584*, 2019. [2](#)
- [28] Roshanak Mirzaee, Hossein Rajaby Faghihi, Qiang Ning, and Parisa Kordjmeshidi. Spartqa: A textual question answering benchmark for spatial reasoning. *arXiv preprint arXiv:2104.05832*, 2021. [2](#)
- [29] Thanh-Son Nguyen, Hong Yang, Tzeh Yuan Neoh, Hao Zhang, Ee Yeo Keat, and Basura Fernando. Neuro symbolic knowledge reasoning for procedural video question answering. *arXiv preprint arXiv:2503.14957*, 2025. [3](#)
- [30] Matthias Plappert, Christian Mandery, and Tamim Asfour. The kit motion-language dataset. *Big data*, 4(4):236–252, 2016. [2](#)
- [31] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, 2021. [7](#)
- [32] Jiankai Sun, Bolei Zhou, Michael J Black, and Arjun Chandrasekaran. Locate: End-to-end localization of actions in 3d with transformers. *arXiv preprint arXiv:2203.10719*, 2022. [2](#)
- [33] Guy Tevet, Brian Gordon, Amir Hertz, Amit H Bermano, and Daniel Cohen-Or. Motionclip: Exposing human motion generation to clip space. In *European Conference on Computer Vision*, pages 358–374. Springer, 2022. [1](#), [2](#)
- [34] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *Proceedings of the AAAI conference on artificial intelligence*, 2018. [2](#)
- [35] Rui Yao, Guosheng Lin, Qinfeng Shi, and Damith C Ranasinghe. Efficient dense labelling of human activity sequences from wearables using fully convolutional networks. *Pattern Recognition*, 78:252–266, 2018. [2](#)
- [36] Michihiro Yasunaga, Hongyu Ren, Antoine Bosselut, Percy Liang, and Jure Leskovec. Qa-gnn: Reasoning with language models and knowledge graphs for question answering. *arXiv preprint arXiv:2104.06378*, 2021. [2](#)
- [37] Kexin Yi, Jiajun Wu, Chuang Gan, Antonio Torralba, Pushmeet Kohli, and Josh Tenenbaum. Neural-symbolic vqa: Disentangling reasoning from vision and language understanding. *Advances in neural information processing systems*, 31, 2018. [2](#), [3](#)
- [38] Qing Yu, Mikihiro Tanaka, and Kent Fujiwara. Exploring vision transformers for 3d human motion-language models with motion patches. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 937–946, 2024. [2](#), [4](#), [5](#)
- [39] Mingyuan Zhang, Huirong Li, Zhongang Cai, Jiawei Ren, Lei Yang, and Ziwei Liu. Finemogen: Fine-grained spatio-temporal motion generation and editing. *Advances in Neural Information Processing Systems*, 36:13981–13992, 2023. [2](#), [6](#)
- [40] Yuanhan Zhang, Jinming Wu, Wei Li, Bo Li, Zejun Ma, Ziwei Liu, and Chunyuan Li. Video instruction tuning with synthetic data. *arXiv preprint arXiv:2410.02713*, 2024. [2](#)

## Supplementary Material for

# IMoRe: Implicit Program-Guided Reasoning for Human Motion Q&A

**Comparison with video-language models trained on large-scale data.** We finetuned Qwen-2.5-VL-3B, InternVL2.5-4B-MPO and MiniCPM-V2.6 on the Babel-QA dataset where the motion sequences are converted to videos consisting of skeleton images. Program and answer set information are provided in the prompt for fair comparison. As shown in Table 1, IMoRe significantly outperforms all VLM baselines. We attribute this to: (1) VLMs struggle to capture fine-grained temporal and spatial concepts from motion image sequences, and (2) their limited ability to leverage structured programs during reasoning.

	Overall	Action	Direction	Bodypart
IMoRe I	0.609	0.652	0.622	0.373
Qwen-2.5-VL-3B	0.425	0.467	0.333	0.350
InternVL2.5-4B	0.402	0.433	0.306	0.383
MiniCPM	0.384	0.410	0.306	0.367

Table 1. Comparison with VLMs

**Additional ablation study.** (a) Comparison between question and program for text-aware feature. We fuse the motion feature and the text feature to obtain the text-aware feature as described in Sec. 3.3 of the main paper. To verify this design, we show results using programs for text-aware motion feature (IMoRe w Pro) in Table 2. We can see that the performance drops slightly, likely because the question texts better guide token-level attention in ViT-derived motion features, improving the alignment between text and motion. (b) Question type information. We also ablate over the question type information as described in Sec 3.3 of the main paper. We can see that the performance without question type (IMoRe wo QT) in Table 2 slightly drops.

	Overall	Action	Direction	Bodypart
IMoRe I	0.609	0.652	0.622	0.373
IMoRe w Pro	0.598	0.690	0.583	0.217
IMoRe wo QT	0.603	0.655	0.596	0.254

Table 2. Additional ablation study

**Visualization of feature level selection.** We have compared single-level and multi-level feature selection quantitatively in the ablation study (C vs D setting in Table 3 of main paper). The improvement from C to D verifies the effectiveness of the multi-level based feature selection. We further show a

qualitative attention map between program functions (y-axis) and multi-level features (x-axis) in Fig. 1. It can be seen that **filter\_action()** attends to high-level feature (Feat 6) while **query\_body\_part()** to low-level features (Feat 0 and 1).

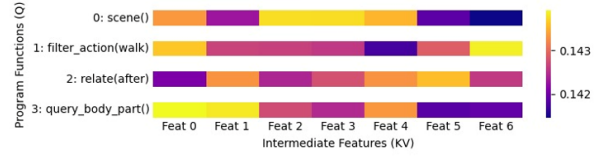


Figure 1. Visualization of feature level selection.

**Visualization of concept localization.** Our approach retains interpretability in two ways despite the implicit program guidance. (1) Program structure: structured programs define an explicit step-by-step reasoning path, where each function (*e.g.*, filter, relate) defines a specific operation. Although the reasoning is implemented implicitly, our model follows this program-defined sequence, making each reasoning step directly attributable to a specific symbolic instruction. (2) Intermediate traceability: As shown in Fig. 2, the attention map between program functions (y-axis) and motion segments (x-axis) reveals interpretable reasoning. After initialization at step 0, **filter\_action(crawl)** at step 1 correctly attends to segment 2 (where the action occurs), thereafter attention for **relate(after)** and **query\_action()** shifts towards segment 3, illustrating stepwise execution consistent with the program.

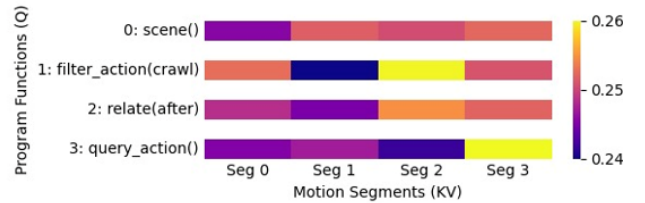


Figure 2. Visualization of concept localization.