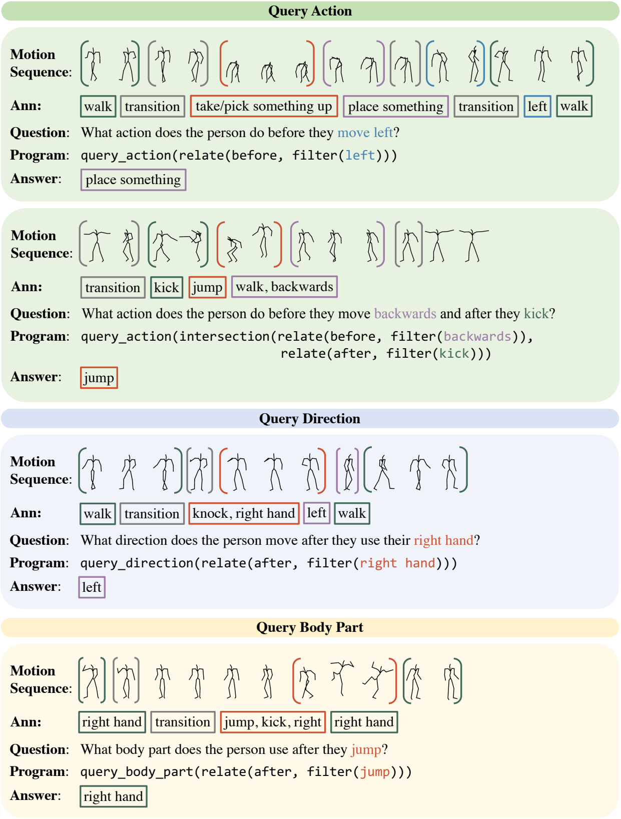
Motion Question Answering via Modular Motion Programs

Mark Endo [*1] **Joy Hsu** [*1] **Jiaman Li** [1] **Jiajun Wu** [1]

Abstract

In order to build artificial intelligence systems that can perceive and reason with human behavior in the real world, we must first design models that conduct complex spatio-temporal reasoning over motion sequences. Moving towards this goal, we propose the HumanMotionQA task to evaluate complex, multi-step reasoning abilities of models on long-form human motion sequences. We generate a dataset of question-answer pairs that require detecting motor cues in small portions of motion sequences, reasoning temporally about when events occur, and querying specific motion attributes. In addition, we propose NSPose, a neuro-symbolic method for this task that uses symbolic reasoning and a modular design to ground motion through learning motion concepts, attribute neural operators, and temporal relations. We demonstrate the suitability of NSPose for the HumanMotionQA task, outperforming all baseline methods.



Question: What action does the person do before they move left?

Program: `query_action(relate(before, filter(left)))`

Answer: place something

Question: What action does the person do before they move backwards and after they kick?

Program:
```
query_action(intersection(relate(before, filter(backwards)),
                          relate(after, filter(kick)))
```

Answer: jump

Question: What direction does the person move after they use their right hand?

Program: `query_direction(relate(after, filter(right hand)))`

Answer: left

Question: What body part does the person use after they jump?

Program: `query_body_part(relate(after, filter(jump)))`

Answer: right hand

Figure 1. For the task of human motion question answering (HumanMotionQA), we create a dataset (BABEL-QA) that evaluates models' ability to learn complex, multi-step reasoning for human behavior understanding. We present examples of several types of questions in our dataset, including querying for action, direction, and body part across temporal relations.

1. Introduction

A longstanding research goal in artificial intelligence is to build models that can perceive and interact with humans in the real world. To achieve this goal, we must first understand complex human behavior across space and time; hence, we are interested in the characterization of long-form human motion sequences in 3D scenes. The growing amount of available human motion capture data in recent years has enabled the development of a variety of tasks (Mahmood et al., 2019; Shahroudy et al., 2016; Punnakkal et al., 2021), including action recognition (Caba Heilbron et al., 2015), motion forecasting (Martínez-González et al., 2021), and temporal localization (Sedmidubsky et al., 2019). Although these tasks involve the understanding of motion sequences, none require complex, multi-step reasoning about both action-level events (e.g., how behaviors are performed and relate to one another) as well as frame-level fine-grained detection (e.g., body parts involved in specific frames and sudden changes of direction).

Thus, we propose the task of human motion question answering, HumanMotionQA, to evaluate such complex and fine-grained human behavior understanding (See Figure 1). Our task consists of a human motion sequence, paired with a question in natural language and an answer from a vocabulary of words. The questions pertain to different attributes

[*] Equal contribution [1] Department of Computer Science, Stanford University. Correspondence to: Mark Endo <markendo@stanford.edu>.

Proceedings of the 40th International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023.

in the motion sequences such as action, direction, and body part, and involve temporal relations such as before, after, and in between. HumanMotionQA requires complex motion understanding and spatio-temporal reasoning, as models must (1) detect subtle and complex motor cues performed only in a small portion of a motion sequence and (2) reason temporally about how different sections in a motion sequence relate to one another without having access to action boundaries. To explore the task of HumanMotionQA, we build a dataset BABEL-QA based on BABEL (Punnakkal et al., 2021) and AMASS (Mahmood et al., 2019). BABEL-QA comprises 1109 motion sequences with 2577 associated question-answer pairs and is an important step to understanding complex human behavior.

Learning a mapping of human motions and questions to corresponding answers is challenging for two key reasons. First, complex motion reasoning requires grounding different actions in untrimmed motion sequences without access to explicit action boundaries. Second, models typically require large amounts of data and suffer from data biases such as imbalanced action co-occurrences. To enable explicit grounding in untrimmed motion, we propose to decompose the untrimmed sequence into overlapped motion segments so that we can model the relationship between each segment and the question. Moreover, we adopt a neuro-symbolic framework to eliminate the need for large-scale data and mitigate potential data biases. Our proposed approach, NSPose, executes symbolic programs recursively on the input motion sequence and learns modular motion programs that correspond to different activity classification tasks. Our method jointly learns motion representations and language concept embeddings from motion sequences and question-answer pairs. Compared to end-to-end approaches applied to the HumanMotionQA task, NSPose enables improved temporal grounding capabilities. By leveraging the program structure specified in language, we achieve effective learning of human motion concepts (e.g. activities such as *walking* and *jumping*, activity characteristics such as *forward* and *backward*, and body parts such as *left arm* and *right leg*), leading to a faithful grounding of human trajectories in motion sequences.

We show that NSPose results in improved question-answering performance compared to baseline end-to-end methods for the task of HumanMotionQA. Our method is capable of complex, multi-step reasoning by using decomposed program structures to learn modular human motion concepts. Importantly, NSPose learns temporal grounding without action localization supervision, resolving prior neuro-symbolic visual reasoning approaches' need for ground truth segments. In summary, we jointly propose BABEL-QA, a new dataset for human motion question answering, as well as NSPose, a neuro-symbolic solution designed for this task. Both extend current deep learning capabilities for human behavior understanding.

2. Related Work

Motion reasoning. In recent years, action recognition for human motion has been extensively studied (Yan et al., 2018; Asghari-Esfeden et al., 2020; Caetano et al., 2019; Cai et al., 2021; Chen et al., 2021a; Cheng et al., 2020; Choutas et al., 2018; Du et al., 2015; Ke et al., 2017; Liu et al., 2020; Shi et al., 2019; 2020). Leading approaches such as ST-GCN (Yan et al., 2018) used a graph convolution model to capture the spatial-temporal relationship among joints in different time steps. A typical research paradigm has been focused on designing robust GCN-based model architectures to improve action recognition accuracy given a sequence of joint positions. Recently, PoseConv3D (Duan et al., 2022) revisited pose representation for the action recognition task and proposed a 3D heatmap volume representation to utilize a powerful 3D-CNN model, leading to superior results compared to previous approaches. Skeleton-based action recognition requires trimmed motion segments as input to estimate the probability of action labels. To predict action labels from untrimmed motion sequences, temporal convolution network (Filtjens et al., 2022; Yao et al., 2018) and transformer model (Sun et al., 2022) was adopted to estimate per-frame action probability so that the action localization task can be accomplished by aggregating per-frame predictions. However, these works rely on expensive temporal annotations for action segments and are incapable of providing a fine-grained understanding of long motion sequences that require multi-step reasoning. In this work, we aim to ground the actions without the need for temporal action annotations and address the task of human motion question-answering for complex reasoning on human behaviors.

Joint learning of motion and language. Prior work on skeleton-based recognition and localization learned neural models from datasets consisting of paired motion and action labels (Liu et al., 2017; Chereshnev & Kertész-Farkas, 2018; Niemann et al., 2020). However, a human motion sequence conveys more than a single action label. We can recognize the moving direction of a walking sequence, perceive the body parts involved in each action and infer the temporal relationships between actions. To provide a detailed description of human motion, recent datasets (Punnakkal et al., 2021) annotated natural language on top of the existing motion capture datasets (Mahmood et al., 2019) to facilitate the joint modeling of motion and language. These datasets have led to growing research on generating human motions from language descriptions (Guo et al., 2022; Athanasiou et al., 2022; Tevet et al., 2022; Petrovich et al., 2022; Zhang et al., 2022; Kim et al., 2022). For example, conditional VAE was adopted to generate natural human movements conditioned on text (Guo et al., 2022). Recently, with the success of the diffusion model in various generative tasks, motion generation results have been greatly improved by applying the diffusion formulation to human motion (Zhang et al., 2022; Kim et al., 2022). Though the generative task from text has been widely studied based on the datasets with motion and language modalities, the motion recognition and reasoning tasks were neglected in the literature. We propose a motion question-answering task for fine-grained motion understanding in this work.

Neuro-symbolic approaches. Neuro-symbolic approaches have proven to be successful in visual reasoning tasks (Yi et al., 2018; Mao et al., 2019). Neuro-symbolic VQA (Yi et al., 2018) combined symbolic program execution and visual recognition to address the question-answering task, leading to superior performance in the CLEVR benchmark (Johnson et al., 2017). NS-CL (Mao et al., 2019) further eliminated the need for dense supervision and designed an effective paradigm to train the neuro-symbolic module by looking at images and reading questions and answers. Recently, neuro-symbolic frameworks have also been extended to temporal reasoning tasks (Chen et al., 2021b) and 3D reasoning problems (Hong et al., 2022; Hsu et al., 2023), showcasing the capability of grounding concepts with weak supervision and generalizing to new language compositions. Inspired by the success of neuro-symbolic approaches in various tasks, we devise a neuro-symbolic framework for motion sequences to address the task of human motion question-answering with natural supervision (questions and answers). By leveraging paired motion and question-answer pairs, we can ground actions concepts temporally, reason about the temporal relations of action segments, and infer attributes such as the moving direction and the body parts involved in each action.

3. HumanMotionQA and BABEL-QA

For the HumanMotionQA task, we introduce the BABEL-QA dataset, which consists of human motion sequences paired with questions in natural language and answers from a vocabulary of words. We describe the task in Section 3.1 and the dataset details in Section 3.2.

3.1. The HumanMotionQA task

Given a sequence of human motion capture data represented with 3D joint positions, $\mathcal{S} \in \mathbb{R}^{T\times J\times 3}$, where T is the number of timesteps in the motion sequence and J is the number of joints, and a question about the sequence, the goal of HumanMotionQA is to predict the corresponding answer by reasoning about the motion sequence $\mathcal{S}$. Each motion sequence $\mathcal{S}$ consists of a temporal composition of several human actions chained together sequentially. For example, a motion sequence can comprise a person kicking a ball with their left foot, running forward, then jumping. For our task, an example corresponding question is "What direction does the person move before jumping and after using their left foot?" For a model to reliably answer this question correctly, it must first understand where in the sequence the person is jumping and using their left foot, understand the time period between these two events, and know what direction they are moving in that time frame. Questions in BABEL-QA require multi-step reasoning – encompassing human motion classification, attribute-specific queries, and an understanding of temporal relations.

The HumanMotionQA task evaluates how well models can detect subtle motor cues performed on only a portion of long-form motion sequences, and the multi-step reasoning abilities of models to first detect motor cues, then reason temporally about action boundaries, and lastly query attributes relating to actions, direction, and body parts.

3.2. The BABEL-QA dataset

To build BABEL-QA, we create question-answer pairs from motion sequences and annotations in the BABEL dataset (Punnakkal et al., 2021). We leverage BABEL, as it contains dense labels that describe each individual action in the temporal composition, in addition to when the action occurs in the motion sequence. This dense information allows us to extract motion concepts from discrete parts of the motion sequences and procedurally build questions by processing temporal relations.

The questions in our dataset relate to three categories of motion attributes: action, direction, and body part. Each attribute contains various concepts such as *walk* and *run* for the action attribute, *forward* and *backward* for the direction attribute, and *right arm* and *left leg* for the body part attribute. To compose questions that require reasoning about these different concepts, we use the following logical building blocks: `filter`, `relate`, and `query`. The `filter` function selects the subset of motion segments that contain a certain concept. The `relate` function selects a subset of motion segments that satisfy a certain temporal relation. For example, if you apply a *before* relation to a segment, the function selects the preceding segment. The `query` function outputs what concept is contained in a motion segment for an attribute of interest.

Our questions follow the structure of first filtering for a concept, optionally applying temporal relation(s), then querying for an attribute. For example, given a sequence of someone throwing a ball with their right hand and then running, we can create a question to first `filter` for the *run* motion, then add a temporal `relate` function for the *before* relation, and finally `query` for the body part. In natural language form, this question is "What body part does the person use before they run?" With this question structure, we have three different question types, each categorized by the attribute for the `query` function. Within each question type, we also categorize sub-question types according to the intermediate relation function (either *before*, *after*, *in between*, or no temporal relation).

To create question-answer pairs from the BABEL dataset, we first extract motion concepts from the sequences by parsing frame-level label texts and action categories. To avoid creating ambiguous questions, we remove action categories that can contain many different types of movements (e.g., animal behavior). Using these extracted motion concepts with temporal ordering, we then sequentially construct questions with our function building blocks. For each unique concept in the motion sequence (only existing in one segment of the temporal composition), we create new sets of questions by filtering for that segment's concept. We then procedurally generate various types of questions building on this first operation by applying possible temporal relations.

If the segment that immediately precedes the filter segment has extracted motion concepts, then we add a *before* relation and create a query question for each annotated attribute in that segment (e.g., action, direction, and/or body part). Likewise, if the segment that immediately follows the filter segment has an extracted motion concept, then we add an *after* relation and create query questions for each attribute. Note that in the case where the segment immediately preceding or following the filter segment is annotated with the *transition* action, we ignore the segment and look one segment before or after for temporal relations. We can also create questions with both *before* and *after* relations (*in between*) by additionally filtering for a concept for the segment on the other side of the query segment, applying the opposite temporal relation, and combining the two relation outputs with `intersection` before querying. Lastly, if the filter segment contains additional extracted motion concepts, then we create query questions for each additional attribute without the use of temporal relations. For the BABEL train, validation, and test splits, we generate every possible question in this format and remove questions with concepts that appear less than eight times.

As BABEL consists of natural human motion sequences, certain concepts often occur together either in the same motion segment or adjacent segments. This concurrence of action characteristics causes data bias in co-occurrences between filter concepts and query attribute answers, which systems can easily exploit to answer questions without learning the underlying reasoning process. For example, the answer to the question "What action does the person do before standing up?" will often be "sit down". To solve this issue, we downsample questions that have common co-occurences. Specifically, given a filter concept c_i and query attribute a_k,

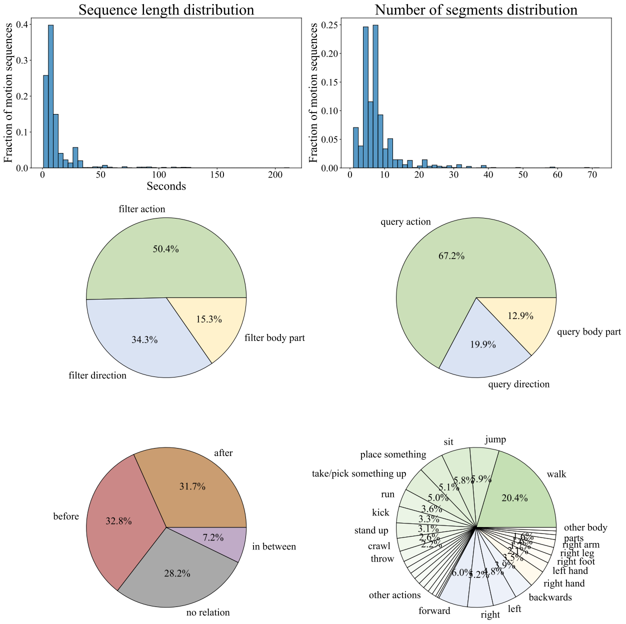


Figure 2. **Top:** distribution of motion sequence length and number of segments (discrete actions) in motion sequences. **Bottom:** distribution of filter types, query types, temporal relation types, and query answers

we count the number of times each answer α_j occurs when first filtering for c_i then querying on a_k (noted as $c_i \to a_k$). We then balance the dataset such that

$$\frac{\texttt{Count}(\alpha_j)}{\sum_{l \in \text{answers for } c_i \to a_k} \texttt{Count}(\alpha_l)} < \tau$$

for all $j \in$ answers for $c_i \to a_k$, where τ is a threshold set at 34%.

With this processing, our final dataset is composed of 771 train motion sequences, 167 validation motion sequences, and 171 test motion sequences with an associated 1800 train questions, 384 validation questions, and 393 test questions. Figure 2 contains information about data statistics. The code for generating this dataset is available at `https://github.com/markendo/HumanMotionQA/`. Additional details on the BABEL-QA dataset and the labeling process can be found in the Appendix.

We propose HumanMotionQA and BABEL-QA to evaluate complex reasoning on real-world human motion. As our dataset comes from BABEL, it contains real-world human motion capture of many types of movements. In addition, our dataset is not limited to joint positions as input. BABEL-QA provides joint position and rotation representations, as well as full body and hand meshes. Importantly, our dataset contains examples sampled from BABEL, which contains different types of actions and a large variation in the com-

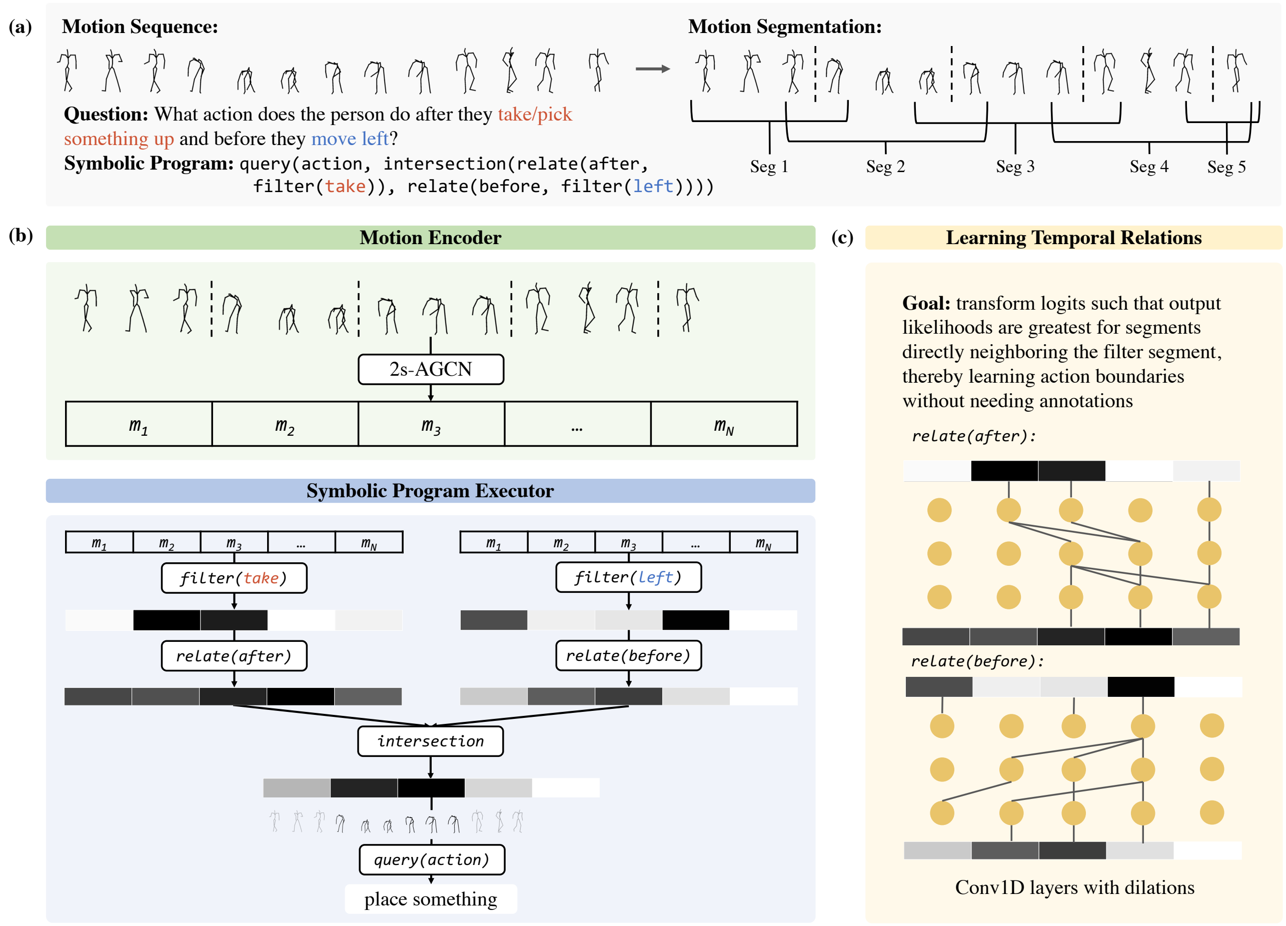

(a)
Motion Sequence:
Motion Segmentation:
Question: What action does the person do after they take/pick something up and before they move left?
Symbolic Program: query(action, intersection(relate(after, filter(take)), relate(before, filter(left))))
Seg 1
Seg 2
Seg 3
Seg 4
Seg 5
(b)
Motion Encoder
2s-AGCN
m_1
m_2
m_3
...
m_N
Symbolic Program Executor
filter(take)
filter(left)
relate(after)
relate(before)
intersection
query(action)
place something
(c)
Learning Temporal Relations
Goal: transform logits such that output likelihoods are greatest for segments directly neighboring the filter segment, thereby learning action boundaries without needing annotations
relate(after):
relate(before):
Conv1D layers with dilations

Figure 3. Framework of NSPose. **(a)** Overview of extracting motion frames from long-form human motion capture data and the symbolic program structure of BABEL-QA questions. **(b)** Visualization of motion feature extraction and program execution using `filter`, `relate`, and `query` functions. **(c)** Approach of learning relations which are required for the model's temporal understanding and multi-step reasoning abilities.

position of motions. Each question is complex and requires reasoning about many aspects of the motion. With these different components, we can test methods' performance on complex real-world reasoning on real-world data.

4. Methods

In Section 4.1, we present NSPose, a neuro-symbolic method that we developed to solve the HumanMotionQA task. In Section 4.2, we discuss additional baselines we explore for question answering in human motion sequences.

4.1. NSPose

We introduce NSPose as a method that leverages a symbolic reasoning process to learn motor cues, modular concepts relating to motion (actions, directions, and body parts), and temporal relations. NSPose takes as input a human motion sequence as well as an executable program and outputs an answer from a vocabulary of words. We give an overview in Figure 3.

In Figure 3 (a), our method first splits the input motion human sequence into N segments. We create overlapping segments of a set frame length such that each segment captures a distinct part of the full sequence with surrounding motion context.

Then, in Figure 3 (b), NSPose learns motion encodings for each segment, resulting in modular representations $m_1, ..., m_N$ that span the full motion sequence. Finally, NSPose recursively executes the program trace with motion representations, jointly learning motion concept embeddings and temporal relation transformations. NSPose's programs are executed as neural networks; in Figure 3 (c), the temporal transformation program is implemented as 1D convolutional layers with dilation, enabling learning of temporal action boundaries. The program executor is fully differentiable with respect to the motion representations and concept embeddings, which allows for gradient-based optimization.

NSPose improves prior work in neuro-symbolic reasoning in two main ways. The first is the handling of variable length temporal motion sequences, compared to 2D images. We train NSPose to recognize complex human motion with a skeleton-based feature extraction. The second is NSPose 's joint learning of action localization and the downstream question answering task. Prior neuro-symbolic visual reasoning approaches such as NS-CL require object-centric input (e.g., object bounding boxes, or translated to our temporal domain, action segments) (Mao et al., 2019). NSPose does this learning jointly through a temporal projection layer, trained in conjunction with the motion feature extractor and the program executor. We detail each part of NSPose below.

Motion feature extractor. We use a Two-Stream Adaptive Graph Convolutional Network (2s-AGCN) model to encode motion segments $\mathcal{S}_1, ..., \mathcal{S}_N$ into embedded motion features $m_1, ..., m_N$ (Shi et al., 2019). This model goes beyond the conventional GCN approach for skeletal-based action recognition (Yan et al., 2018) of using a predefined human-body-based graph and instead parameterizes two learned types of graphs. This adaptation increases the flexibility of the model and allows the model to learn different human graph structures for different types of activities.

Notably, NSPose operates on full motion sequences, without requiring ground truth action boundaries. We split each input motion sequence into segments of f frames, with varying number of segments in each sequence. We also overlap segments by o frames on each side in order to provide the model with more context in each segment. In our experiments, we set $f = 45$ and $o = 15$. NSPose's motion feature extractor operates on these frame segmentations, and learns to ground each to a motion concept or attribute. Our method is tasked with action localization in order to answer questions involving temporal operations, while solely supervised by questions and answers in natural language, without pre-training the 2s-AGCN motion encoder.

Neuro-symbolic framework. To answer questions that involve multi-step reasoning about complex activity characteristics across space and time, we propose NSPose as a neuro-symbolic framework. We extend prior neuro-symbolic visual reasoning methods (Mao et al., 2019), which operates on 2D images and requires object segmentations, to NSPose, which operates on motion sequences and can learn temporal grounding of frames to action concepts without segmentations of action boundaries. We detail NSPose's program executor below.

First, let us denote A as the set of all motion attributes (e.g., *action*, *direction*, and *body part*) and C as the set of all concepts (e.g., *walk*, *forward*, *left foot*, etc.). For each motion concept $c \in C$, we learn a vector embedding v^c that represents this concept. We also learn an L1-normalized vector b^c that represents the likelihood of the concept belonging to each of the attributes. In addition, we learn neural operators for each attribute $a \in A$ as u^a that transform motion features to the a attribute embedding space.

With these embeddings, vectors, and neural operators, we define the `filter` and `query` programs. The `filter` function takes as input the motion segment embeddings $m_1, ..., m_N$ and a concept of interest c (e.g., *sit*) and returns logits for which segments are most likely to contain the input concept. For a single segment embedding m_i, we first calculate the likelihood that m_i includes c as

$$\sigma\left(\sum_{a \in A}\left(b_a^c \cdot \frac{\langle u^a(m_i), v^c\rangle - \gamma}{\tau}\right)\right),$$

where σ is the Sigmoid function, $\langle \cdot, \cdot \rangle$ is cosine distance, and γ and τ are scalar constants. In the `filter` operation, we calculate this likelihood, which we shorten as `motion_classify`(m_i, c), for every motion segment.

For the `query` function, we query an attribute on the motion segments using input segment weights $w_1, ..., w_N$ which are logits returned by either the `filter` or `relate` function. We similarly define the likelihood that the input belongs to a concept c as

$$\sum_{i=1}^{N} w_i \cdot \frac{\texttt{motion\_classify}(m_i, c) \cdot b_a^c}{\sum_{c' \in C} \texttt{motion\_classify}(m_i, c') \cdot b_a^{c'}}.$$

We calculate this likelihood p_c for every concept and define the loss as $-\log \frac{\exp(p_y)}{\sum_{c \in C} \exp(p_c)}$, where y is the ground truth concept.

Temporal grounding. In addition to learning motion concepts and transformations from the motion to attribute embedding space, we also learn `relate` operators that capture temporal relations for *before*, *after*, and *in between* from human motion frames, without the use of annotated action boundaries. The `relate` functions take in motion segment logits and transform the logits according to the temporal relation of interest, learning action boundaries for the input motion sequence. To learn these temporal transformations, we leverage a convolutional neural network model consisting of 1D convolutional layers with dilation, which has been proven to be successful for learning motifs in sequential data (Avsec et al., 2021).

Given the input segment weight vector $W = [w_1, ..., w_N]$ from the preceding filter function, we return CNN(W), where CNN has three intermediate convolution layers with 16 filters per layer, kernel size of three, and and exponential dilation in every layer. We additionally explore a baseline approach of using a simple linear layer that translates the vector logits to another vector of transformed logits. Though NSPose is trained with only a final answer cross entropy loss, without any intermediate losses, it is able to learn temporal grounding of frames to action concepts through question answering pairs in natural language as weak supervision.

NSPose is able to identify boundaries between different actions, as these transition frames are learned implicitly through filtering for concepts in segments with temporal relations. We show qualitative results of NSPose 's temporal grounding capabilities in Figure 4. Although the predicted boundaries of our model accurately capture transitions, one constraint of these boundaries is that they are predicted at the segment level instead of the model predicting a specific timepoint. To make the boundaries more exact, it is possible to create more segments per motion sequence by reducing the number of frames in each segment. However, the drawback of this change is that there would be less motion context in each segment for the motion encoder to learn from. Through experimentation, we found that having 45 frame segments with 15 frames of overlap is a good balance between having large enough segments to learn useful motion cues and having small enough segments to have fine-grain boundary predictions.

4.2. Baselines

We compare our method against five different baselines. The first baseline uses only question text to answer questions, resulting in a model that can only exploit possible data bias. The second two baselines are built upon a recent method for learning powerful human motion latent representations (Tevet et al., 2022). The last two baselines are end-to-end methods that leverage question text and the same skeleton-based feature extractor we use in our approach (Shi et al., 2019).

CLIP. This method solely uses the question texts and not the motion sequences that are necessary to faithfully answer the corresponding questions. Specifically, we pass the questions into to a pre-trained CLIP model (Radford et al., 2021) to get text embeddings and then train a simple multilayer perceptron (MLP) on top to predict question answers. We use this method as a rudimentary baseline that can only learn text questions and dataset biases.

MotionCLIP-MLP. In this method, we embed both the natural language questions and motion sequences into the same latent representation space such that the two modalities of data can be easily used together for prediction. To do this, we utilize MotionCLIP, a transformer-based motion auto-encoder trained to reconstruct motion while being aligned to its corresponding text's position in the CLIP space (Tevet et al., 2022). We pass the entire motion sequence into the model to attain a single motion representation, and we concatenate this information with the CLIP embedding of the question. We then train an MLP on top to predict answers.

MotionCLIP-RNN. For this baseline, we follow a similar setup to MotionCLIP-RNN, except we pass individual action segments into the model instead of the entire motion sequence. This modification results in attaining one representation for each action segment in the sequence. In order to predict the answer, we utilize a recurrent neural network (RNN). Specifically, we first pass the CLIP embedding of the question into the model as the initial hidden state. The latent motion segment representations are then passed sequentially into the model as inputs. We use the final output of the RNN model as the predicted answer to the question. We conjecture that this change from MotionCLIP-MLP to MotionCLIP-RNN will enable this baseline to discern fine-grain details in the motion sequence since each distinct action has its own embedding. The appendix contains visualizations for the MotionCLIP baseline architectures. For both MotionCLIP baselines, we fine-tune the human motion encoder on our dataset while using frozen CLIP weights.

2s-AGCN-MLP. This baseline is an end-to-end approach that leverages 2s-AGCN to extract motion features. 2s-AGCN-MLP uses the same feature extractor as NSPose, and hence evaluates the importance of modular programs from the symbolic components of NSPose compared to prior end-to-end regimes. Concatenating a CLIP embedding of the question with a single 2s-AGCN motion representation, we train an MLP on top to predict answers. Similarly to MotionCLIP-MLP, we fine-tune the human motion encoder.

2s-AGCN-RNN. In this setup, we use the same motion feature encoder, 2s-AGCN, but utilize a recurrent neural network (RNN) to predict the answer. We follow the same prediction process as MotionCLIP-RNN but use motion embeddings from 2s-AGCN instead of MotionCLIP.

5. Experiments

We investigate the performance of NSPose and baseline methods on the BABEL-QA test set. Table 1 contains detailed results of all methods. We compare NSPose to baseline methods in Section 5.1 and present ablations of NSPose in Section 5.2.

5.1. Comparison to baselines

Our findings show that NSPose outperforms all the baseline methods in overall accuracy. Notably, our method outperforms the deeper MotionCLIP baselines, which are pre-trained on the BABEL dataset to learn CLIP-aligned human motion latent representations. Our method has an overall

Table 1. Evaluation of NSPose and baseline methods on the BABEL-QA test set. Performance is evaluated using accuracy and we report the mean score of three runs. We find that NSPose performs better than baselines. BTW stands for *in between*.

MODEL	OVERALL	QUERY ACTION				QUERY DIRECTION				QUERY BODY PART			
		ALL	BEFORE	AFTER	BTW	ALL	BEFORE	AFTER	BTW	ALL	BEFORE	AFTER	BTW
CLIP	0.417	0.467	0.380	0.452	0.591	0.366	**0.467**	0.292	0.222	0.261	0.261	0.278	**0.333**
2S-AGCN-MLP	0.355	0.384	0.353	0.411	0.273	0.352	0.378	0.250	0.278	0.228	0.261	0.130	**0.333**
2S-AGCN-RNN	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	0.430	0.485	0.411	0.470	0.545	0.361	0.400	0.271	0.333	0.272	0.304	0.222	**0.333**
MOTIONCLIP-RNN	0.420	0.489	0.461	0.441	0.606	0.310	0.400	0.333	0.222	0.250	**0.333**	0.167	**0.333**
NS-POSE (OURS)	**0.578**	**0.627**	**0.618**	**0.620**	**0.639**	**0.598**	0.389	**0.583**	**0.750**	**0.325**	0.296	**0.471**	0.083

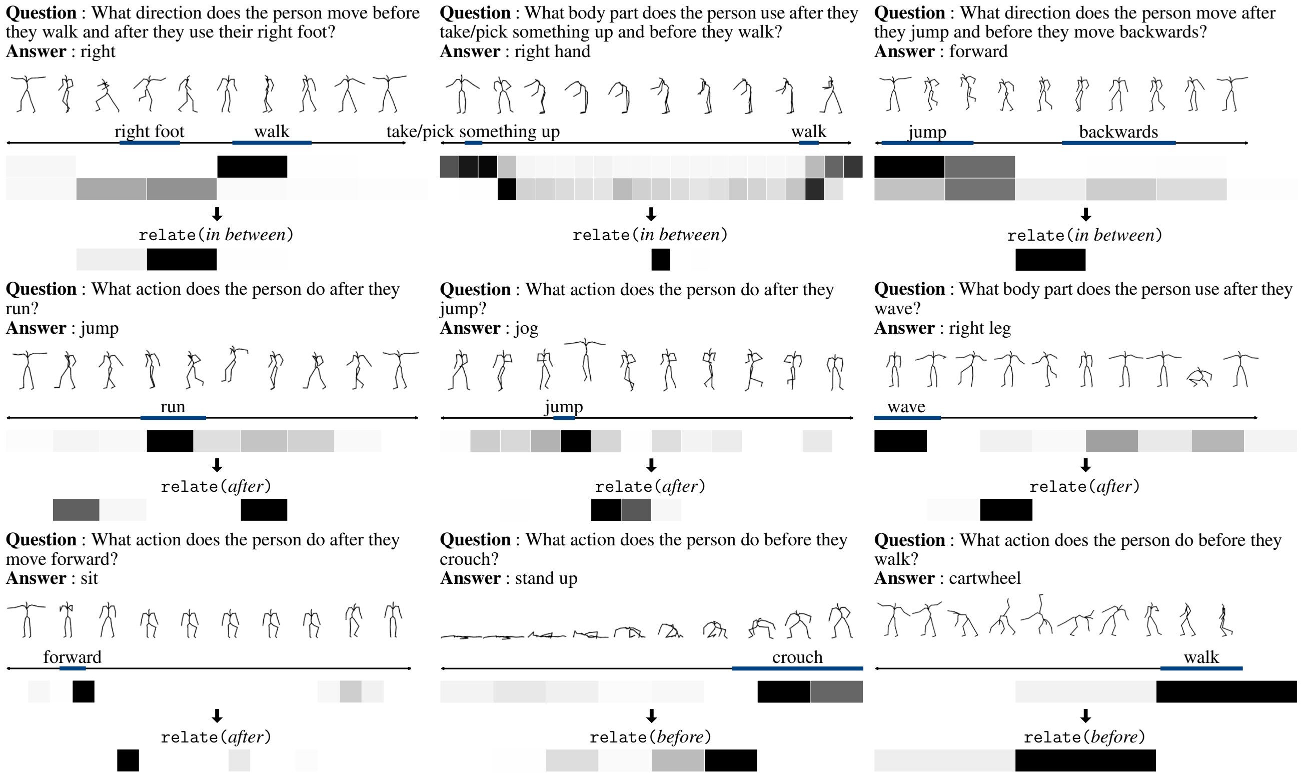

Question : What direction does the person move before they walk and after they use their right foot?
Answer : right
right foot
walk
relate(in between)
Question : What body part does the person use after they take/pick something up and before they walk?
Answer : right hand
take/pick something up
walk
relate(in between)
Question : What direction does the person move after they jump and before they move backwards?
Answer : forward
jump
backwards
relate(in between)
Question : What action does the person do after they run?
Answer : jump
run
relate(after)
Question : What action does the person do after they jump?
Answer : jog
jump
relate(after)
Question : What body part does the person use after they wave?
Answer : right leg
wave
relate(after)
Question : What action does the person do after they move forward?
Answer : sit
forward
relate(after)
Question : What action does the person do before they crouch?
Answer : stand up
crouch
relate(before)
Question : What action does the person do before they walk?
Answer : cartwheel
walk
relate(before)

Figure 4. Visualization of NSPose's temporal grounding results from weak supervision of question-answer pairs only. For each motion sequence, we present the corresponding question, answer, and motion sequence, along with ground truth action boundaries, as well as predicted boundaries by NSPose and outputs after temporal `relate` operators. The rows of rectangles above the `relate` function represent model outputs for how likely each motion segment contains the filtered concept, where darker-colored squares signify a higher likelihood. The rows of rectangles below the `relate` function represent model outputs for which segments satisfy the temporal relationship (come directly before, after, or in between the filtered segments). For sequences with two rows of rectangles above the relation function, the two rows represent outputs for the two filtered concepts with the concept appearing first in the question text on top. Note that for visualization purposes, we use the variant of NSPose without overlapping motion segments.

performance improvement over CLIP by 0.161, an improvement over MotionCLIP-MLP by 0.148, and an improvement over MotionCLIP-RNN by 0.158. NSPose also significantly outperforms both end-to-end 2s-AGCN baselines. The relatively low performance of 2s-AGCN-MLP and 2s-AGCN-RNN shows that our method does not owe its success to the 2s-AGCN motion feature extractor.

We conjecture that the improved performance of NSPose is due to our neuro-symbolic approach that learns modular programs. Instead of exploiting data bias during training, our model learns to ground individual motion concepts and can be accurately applied to the validation set with different compositions of motion concepts. We present full results comparing different methods in Table 1.

Table 2. Ablations of NSPose on the BABEL-QA test set. Performance is evaluated using accuracy and we report the mean score of three runs. QU. stands for `query` and FIL. stands for `filter`.

SEG STRAT	TEMP GROUND	OVERALL	QU. ACTION	FIL. ACTION	QU. DIRECTION	FIL. DIRECTION	QU. BODY PART	FIL. BODY PART
$f + o$	CONV1D	0.578	0.627	0.509	0.598	0.473	0.325	0.454
f	CONV1D	0.540	0.573	0.457	0.577	0.463	0.332	0.424
f	LINEAR	0.540	0.602	0.505	0.548	0.529	0.275	0.495
GT	CONV1D	0.553	0.601	0.620	0.583	0.505	0.271	0.313
GT	LINEAR	0.549	0.606	0.626	0.587	0.599	0.266	0.460

5.2. Ablation studies

We also show ablations with different setups of NSPose. In Section 5.2.1, we compare our method of splitting motion sequences into segments of f frames with o frames of segment overlap, to the approach of not overlapping frames, and a variant that leverages ground truth action boundary annotations to create motion segments. In Section 5.2.2, we examine different temporal relation functions. Table 2 contains the results of the various setups.

5.2.1. MOTION SEGMENTATION STRATEGY

We compare NSPose's weakly-supervised approach of grounding temporal action compositions through segmenting motion sequences into n frame segments with o frames of overlap to (1) a simpler approach without frame overlap, and (2) the more annotation-intensive approach of using ground truth action annotations for creating motion segments (See Table 2).

We find that the frame overlapping approach has an overall performance improvement of 0.038 over the method without frame overlap. We hypothesize that overlapping segments add important motion context for improving representations from the feature extractor while maintaining fine-grain information that comes from having a large number of segments.

Overall, we find that our weakly-supervised approach outperforms the variant of NSPose using ground truth action boundaries by 0.025. This performance difference demonstrates that NSPose can faithfully and accurately reason about complex human behavior across time from full motion sequences. See Figure 4 for examples of NSPose's program execution for temporal relations. We provide additional analyses on NSPose performance in the Appendix.

5.2.2. TEMPORAL RELATION FUNCTION

We present ablations for two different strategies of learning temporal relations. We show experiment results from leveraging our proposed model consisting of 1D convolutional layers with dilation, and experiment results using a simple linear model, for the temporal operator. We find that the convolutional approach has similar overall accuracy as the linear approach. The similar performance between the two methods shows that our framework can accurately learn temporal relation transformations using simple functions.

6. Discussion

In this work, we propose the task of human motion question answering, HumanMotionQA, for human behavior understanding, and propose NSPose as a neuro-symbolic solution for this task. HumanMotionQA evaluates models' ability to conduct complex and fine-grained multi-step reasoning across subtle motor cues in motion sequences. NSPose approaches this task by decomposing questions into program structures that are executed recursively on the input motion sequence, and learns modular programs that correspond to different activity classification tasks. Our method exhibits fine-grain reasoning abilities about complex motions and learns temporal grounding from question answering, leading to improved human behavior understanding.

A limitation of NSPose is its dependency on pre-defined motion programs instead of using a semantic parser to translate natural language questions into programs. We do not learn semantic parsing from text, as our focus is on the temporal grounding of motion sequences. A future direction is the inclusion of a trained semantic parsing module to translate questions into programs, enabling broader applicability of our method.

Acknowledgments. We thank Sumith Kulal for providing valuable feedback on the paper. This work is in part supported by Stanford Institute for Human-Centered Artificial Intelligence (HAI), Stanford Wu Tsai Human Performance Alliance, Toyota Research Institute (TRI), NSF RI #2211258, ONR MURI N00014-22-1-2740, AFOSR YIP FA9550-23-1-0127, Analog Devices, JPMorgan Chase, Meta, and Salesforce.

References

Asghari-Esfeden, S., Sznaier, M., and Camps, O. Dynamic motion representation for human action recognition. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 557–566, 2020.

Athanasiou, N., Petrovich, M., Black, M. J., and Varol, G. Teach: Temporal action composition for 3d humans. In *Proceedings of the International Conference on 3D Vision*, 2022.

Avsec, Ž., Weilert, M., Shrikumar, A., Krueger, S., Alexandari, A., Dalal, K., Fropf, R., McAnany, C., Gagneur, J., Kundaje, A., et al. Base-resolution models of transcription-factor binding reveal soft motif syntax. *Nature Genetics*, 53(3):354–366, 2021.

Caba Heilbron, F., Escorcia, V., Ghanem, B., and Carlos Niebles, J. Activitynet: A large-scale video benchmark for human activity understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 961–970, 2015.

Caetano, C., Sena, J., Brémond, F., Dos Santos, J. A., and Schwartz, W. R. Skelemotion: A new representation of skeleton joint sequences based on motion information for 3d action recognition. In *Proceedings of the IEEE International Conference on Advanced Video and Signal Based Surveillance*, pp. 1–8, 2019.

Cai, J., Jiang, N., Han, X., Jia, K., and Lu, J. 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*, pp. 2735–2744, 2021.

Chen, Y., Zhang, Z., Yuan, C., Li, B., Deng, Y., and Hu, W. Channel-wise topology refinement graph convolution for skeleton-based action recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 13359–13368, 2021a.

Chen, Z., Mao, J., Wu, J., Wong, K.-Y. K., Tenenbaum, J. B., and Gan, C. Grounding physical concepts of objects and events through dynamic visual reasoning. In *Proceedings of the International Conference on Learning Representations*, 2021b.

Cheng, K., Zhang, Y., He, X., Chen, W., Cheng, J., and Lu, H. Skeleton-based action recognition with shift graph convolutional network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 183–192, 2020.

Chereshnev, R. and Kertész-Farkas, A. Hugadb: Human gait database for activity recognition from wearable inertial sensor networks. In *Proceedings of the International Conference on Analysis of Images, Social Networks and Texts*, pp. 131–141. Springer, 2018.

Choutas, V., Weinzaepfel, P., Revaud, J., and Schmid, C. Potion: Pose motion representation for action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7024–7033, 2018.

Du, Y., Wang, W., and Wang, L. Hierarchical recurrent neural network for skeleton based action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1110–1118, 2015.

Duan, H., Zhao, Y., Chen, K., Lin, D., and Dai, B. Revisiting skeleton-based action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2969–2978, 2022.

Filtjens, B., Vanrumste, B., and Slaets, P. Skeleton-based action segmentation with multi-stage spatial-temporal graph convolutional neural networks. *IEEE Transactions on Emerging Topics in Computing*, 2022.

Guo, C., Zou, S., Zuo, X., Wang, S., Ji, W., Li, X., and Cheng, L. Generating diverse and natural 3d human motions from text. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5152–5161, 2022.

Hong, Y., Du, Y., Lin, C., Tenenbaum, J., and Gan, C. 3d concept grounding on neural fields. In *Proceedings of Advances in Neural Information Processing Systems*, 2022.

Hsu, J., Mao, J., and Wu, J. Ns3d: Neuro-symbolic grounding of 3d objects and relations. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2023.

Jiang, Y., Ye, Y., Gopinath, D., Won, J., Winkler, A. W., and Liu, C. K. Transformer inertial poser: real-time human motion reconstruction from sparse imus with simultaneous terrain generation. In *SIGGRAPH Asia 2022 Conference Papers*, pp. 1–9, 2022.

Johnson, J., Hariharan, B., Van Der Maaten, L., Fei-Fei, L., Lawrence Zitnick, C., and Girshick, R. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2901–2910, 2017.

Ke, Q., Bennamoun, M., An, S., Sohel, F., and Boussaid, F. A new representation of skeleton sequences for 3d action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3288–3297, 2017.

Kim, J., Kim, J., and Choi, S. Flame: Free-form language-based motion synthesis & editing. *arXiv preprint arXiv:2209.00349*, 2022.

Liu, C., Hu, Y., Li, Y., Song, S., and Liu, J. 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*, pp. 1–8, 2017.

Liu, Z., Zhang, H., Chen, Z., Wang, Z., and Ouyang, W. Disentangling and unifying graph convolutions for skeleton-based action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 143–152, 2020.

Luo, Z., Hachiuma, R., Yuan, Y., and Kitani, K. Dynamics-regulated kinematic policy for egocentric pose estimation. *Advances in Neural Information Processing Systems*, 34: 25019–25032, 2021.

Mahmood, N., Ghorbani, N., Troje, N. F., Pons-Moll, G., and Black, M. J. Amass: Archive of motion capture as surface shapes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5442–5451, 2019.

Mao, J., Gan, C., Kohli, P., Tenenbaum, J. B., and Wu, J. The neuro-symbolic concept learner: Interpreting scenes, words, and sentences from natural supervision. In *Proceedings of the International Conference on Learning Representations*, 2019.

Martínez-González, A., Villamizar, M., and Odobez, J.-M. Pose transformers (potr): Human motion prediction with non-autoregressive transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2276–2284, 2021.

Niemann, F., Reining, C., Moya Rueda, F., Nair, N. R., Steffens, J. A., Fink, G. A., and Ten Hompel, M. Lara: Creating a dataset for human activity recognition in logistics using semantic attributes. *Sensors*, 20(15):4083, 2020.

Petrovich, M., Black, M. J., and Varol, G. Temos: Generating diverse human motions from textual descriptions. In *Proceedings of the European Conference on Computer Vision*, pp. 480–497. Springer, 2022.

Punnakkal, A. R., Chandrasekaran, A., Athanasiou, N., Quiros-Ramirez, A., and Black, M. J. Babel: bodies, action and behavior with english labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 722–731, 2021.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. Learning transferable visual models from natural language supervision. In *Proceedings of the International Conference on Machine Learning*, pp. 8748–8763, 2021.

Sedmidubsky, J., Elias, P., and Zezula, P. Benchmarking search and annotation in continuous human skeleton sequences. In *Proceedings of the International Conference on Multimedia Retrieval*, pp. 38–42, 2019.

Shahroudy, A., Liu, J., Ng, T.-T., and Wang, G. Ntu rgb+d: A large scale dataset for 3d human activity analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1010–1019, 2016.

Shi, L., Zhang, Y., Cheng, J., and Lu, H. Two-stream adaptive graph convolutional networks for skeleton-based action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12026–12035, 2019.

Shi, L., Zhang, Y., Cheng, J., and Lu, H. Decoupled spatial-temporal attention network for skeleton-based action-gesture recognition. In *Proceedings of the Asian Conference on Computer Vision*, 2020.

Sun, J., Zhou, B., Black, M. J., and Chandrasekaran, A. Locate: End-to-end localization of actions in 3d with transformers. *arXiv preprint arXiv:2203.10719*, 2022.

Tevet, G., Gordon, B., Hertz, A., Bermano, A. H., and Cohen-Or, D. Motionclip: Exposing human motion generation to clip space. In *Proceedings of the European Conference on Computer Vision*, pp. 358–374. Springer, 2022.

Xu, L., Lan, C., Zeng, W., and Lu, C. Skeleton-based mutually assisted interacted object localization and human action recognition. *IEEE Transactions on Multimedia*, 2022.

Yan, S., Xiong, Y., and Lin, D. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *Proceedings of the AAAI conference on Artificial Intelligence*, 2018.

Yao, R., Lin, G., Shi, Q., and Ranasinghe, D. C. Efficient dense labelling of human activity sequences from wearables using fully convolutional networks. *Pattern Recognition*, 78:252–266, 2018.

Yi, K., Wu, J., Gan, C., Torralba, A., Kohli, P., and Tenenbaum, J. Neural-symbolic vqa: Disentangling reasoning from vision and language understanding. In *Proceedings of Advances in Neural Information Processing Systems*, 2018.

Zhang, M., Cai, Z., Pan, L., Hong, F., Guo, X., Yang, L., and Liu, Z. Motiondiffuse: Text-driven human motion generation with diffusion model. *arXiv preprint arXiv:2208.15001*, 2022.

A. Supplementary for Motion Question Answering via Modular Motion Programs

A.1. Domain-specific language & program implementations

We define the domain-specific language (DSL) used for the HumanMotionQA task. Table 3 includes signatures and semantics for all functions, and Table 4 includes implementations for all functions.

Table 3. Operations used in the programs of HumanMotionQA.

Function	Signature	Semantics
`Sequence`	() → SegmentSet	Return all motion segments in the sequence.
`Filter`	(SegmentSet, Concept) → SegmentSet	Filter for motion segments that contain a concept.
`Relate`	(SegmentSet, Relation) → SegmentSet	Outputs segments that satisfy the temporal relationship.
`Query`	(SegmentSet, Attribute) → Concept	Queries the attribute of the SegmentSet.
`Intersection`	(SegmentSet, SegmentSet) → SegmentSet	Outputs the intersection of the two segment sets.

Table 4. Implementations for all functions used in the programs of HumanMotionQA.

Signature	Implementation
$\texttt{Sequence}() \rightarrow y$: SegmentSet	$y_i = 10$, for all $i \in \{1, ..., N\}$
$\texttt{Filter}(x$: SegmentSet, c : Concept) → y : SegmentSet	$y_i = \min(x_i, \texttt{motion\_classify}(m_i, c))$
$\texttt{Relate}(x$: SegmentSet, rel : Relation) → y : SegmentSet	$y = \text{Linear}_{rel}(x)$ or $y = \text{CNN}_{rel}(x)$
$\texttt{Query}(x$: SegmentSet, a : Attribute) → y : Concept	$y = \arg\max_{c \in C} \left(\sum_{i=1}^{N} x_i \cdot \frac{\texttt{motion\_classify}(m_i, c) \cdot b_a^c}{\sum_{c' \in C} \texttt{motion\_classify}(m_i, c') \cdot b_a^{c'}} \right)$
$\texttt{Intersection}(x$: SegmentSet, y : SegmentSet) → z : SegmentSet	$z_i = \min(x_i, y_i)$

A.2. Full results

We report the complete results of all methods and setup for each of three runs in Table 5.

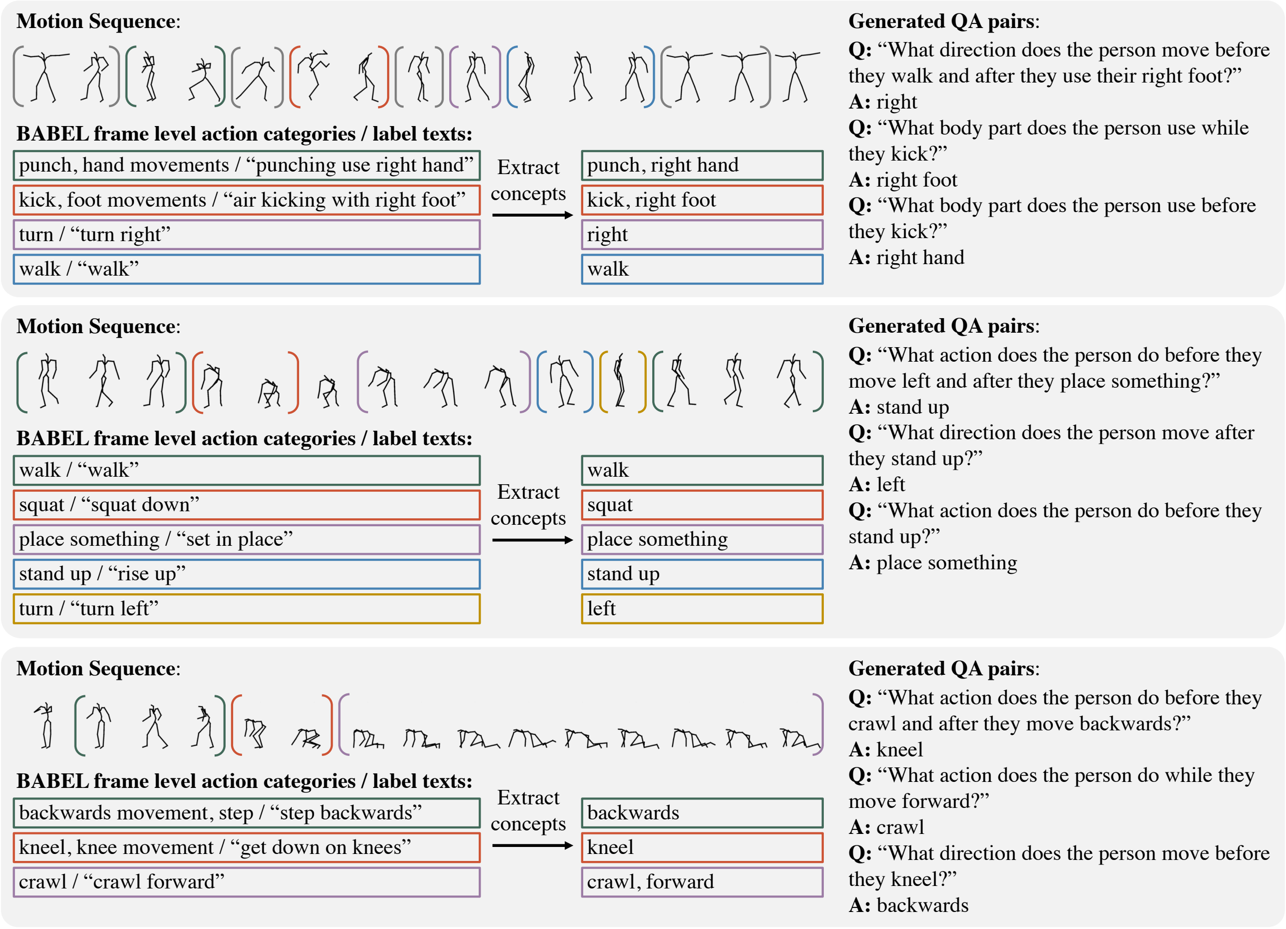
Table 5. Evaluation of various NSPose setups and baseline methods on the BABEL-QA test set. Accuracy is reported for all runs.

Model	Query Action				Query Direction				Query Body Part			
	All	Before	After	Btw	All	Before	After	Btw	All	Before	After	Btw
CLIP	0.456	0.349	0.433	0.591	0.389	0.467	0.375	0.333	0.267	0.304	0.278	0.250
	0.487	0.395	0.478	0.636	0.333	0.467	0.188	0.167	0.267	0.261	0.278	0.500
	0.460	0.395	0.444	0.545	0.375	0.467	0.312	0.167	0.250	0.217	0.278	0.250
2s-AGCN-MLP	0.418	0.360	0.422	0.318	0.361	0.467	0.125	0.500	0.200	0.261	0.056	0.500
	0.398	0.407	0.444	0.318	0.319	0.200	0.312	0.167	0.183	0.174	0.167	0.250
	0.337	0.291	0.367	0.182	0.375	0.467	0.312	0.167	0.300	0.348	0.167	0.250
2s-AGCN-RNN	0.372	0.314	0.400	0.500	0.403	0.467	0.375	0.333	0.200	0.261	0.111	0.250
	0.456	0.419	0.467	0.455	0.306	0.467	0.375	0.167	0.233	0.304	0.167	0.250
	0.360	0.314	0.322	0.273	0.347	0.267	0.438	0.333	0.150	0.217	0.056	0.000
MotionCLIP-MLP	0.487	0.430	0.478	0.455	0.361	0.333	0.312	0.333	0.250	0.217	0.278	0.250
	0.498	0.407	0.500	0.591	0.361	0.467	0.250	0.333	0.267	0.304	0.222	0.250
	0.471	0.395	0.433	0.591	0.361	0.400	0.250	0.333	0.300	0.391	0.167	0.500
MotionCLIP-RNN	0.490	0.453	0.456	0.636	0.375	0.533	0.375	0.167	0.233	0.261	0.167	0.500
	0.502	0.453	0.444	0.591	0.236	0.333	0.188	0.167	0.267	0.348	0.222	0.000
	0.475	0.477	0.422	0.591	0.319	0.333	0.438	0.333	0.250	0.391	0.111	0.500
NS-Pose (f + o, Conv1D)	0.633	0.629	0.648	0.667	0.603	0.417	0.607	0.750	0.321	0.306	0.529	0.000
	0.611	0.589	0.592	0.750	0.603	0.375	0.679	0.750	0.355	0.306	0.412	0.250
	0.636	0.637	0.620	0.500	0.590	0.375	0.464	0.750	0.299	0.278	0.471	0.000
NS-Pose (f, Conv1D)	0.579	0.540	0.570	0.472	0.581	0.375	0.536	0.750	0.359	0.389	0.353	0.500
	0.578	0.556	0.563	0.528	0.587	0.500	0.357	0.750	0.363	0.333	0.176	0.250
	0.561	0.540	0.542	0.444	0.565	0.542	0.429	0.750	0.274	0.389	0.059	0.250
NS-Pose (f, Linear)	0.622	0.621	0.556	0.472	0.609	0.375	0.500	0.750	0.167	0.278	0.118	0.000
	0.593	0.589	0.528	0.528	0.590	0.375	0.429	0.500	0.338	0.278	0.235	0.500
	0.591	0.565	0.507	0.528	0.446	0.125	0.214	0.250	0.321	0.417	0.176	0.500
NS-Pose (GT, Conv1D)	0.637	0.661	0.634	0.556	0.596	0.500	0.357	0.750	0.226	0.333	0.059	0.250
	0.584	0.597	0.563	0.583	0.565	0.458	0.429	0.250	0.316	0.333	0.353	0.250
	0.581	0.532	0.620	0.583	0.587	0.458	0.464	0.750	0.269	0.278	0.176	0.250
NS-Pose (GT, Linear)	0.611	0.589	0.627	0.750	0.593	0.333	0.464	0.750	0.192	0.194	0.265	0.000
	0.596	0.556	0.592	0.417	0.531	0.375	0.143	0.250	0.335	0.306	0.324	0.500
	0.611	0.573	0.599	0.722	0.637	0.500	0.393	0.750	0.269	0.194	0.412	0.000

A.3. Failure mode analyses

We note some areas where models may fail to answer questions from our dataset correctly. One such failure case is when sequences have transition frames between the filter segment and the segment being queried on. For example, in one question the person is moving to the right for 15 frames, transitioning for 20 frames, using their left hand for 22 frames, transitioning for 18 frames, then moving forward for 94 frames. The associated question is "what body part does the person use after they move right and before they move forward?" The periods of transition from one action to the next make the temporal relations less reliable, which will ultimately make the segment weights inaccurate for the query function. Another difficulty with this question is that the person is only using their left hand for 22 frames, which is a very small portion of the overall motion sequence. With the transition periods making the temporal relations difficult and the preciseness needed to pinpoint a body part used in only 22 frames, models are not able to answer this type of question with high accuracy.

We additionally hypothesize that the low performance on query body part questions with between relations is partly due to the fact that encoded motion features don't capture information about body parts very well. Without sufficient information about body location in the embeddings, the learned neural operator for body parts will be ineffective and the transformation from motion features to the body part embedding space will therefore be unreliable. This is supported by the fact that querying body parts is the question type with lowest accuracy across methods.



Motion Sequence:

BABEL frame level action categories / label texts:

punch, hand movements / "punching use right hand"
kick, foot movements / "air kicking with right foot"
turn / "turn right"
walk / "walk"

Extract concepts →

punch, right hand
kick, right foot
right
walk

Generated QA pairs:

Q: "What direction does the person move before they walk and after they use their right foot?"
A: right
Q: "What body part does the person use while they kick?"
A: right foot
Q: "What body part does the person use before they kick?"
A: right hand

Motion Sequence:

BABEL frame level action categories / label texts:

walk / "walk"
squat / "squat down"
place something / "set in place"
stand up / "rise up"
turn / "turn left"

Extract concepts →

walk
squat
place something
stand up
left

Generated QA pairs:

Q: "What action does the person do before they move left and after they place something?"
A: stand up
Q: "What direction does the person move after they stand up?"
A: left
Q: "What action does the person do before they stand up?"
A: place something

Motion Sequence:

BABEL frame level action categories / label texts:

backwards movement, step / "step backwards"
kneel, knee movement / "get down on knees"
crawl / "crawl forward"

Extract concepts →

backwards
kneel
crawl, forward

Generated QA pairs:

Q: "What action does the person do before they crawl and after they move backwards?"
A: kneel
Q: "What action does the person do while they move forward?"
A: crawl
Q: "What direction does the person move before they kneel?"
A: backwards

Figure 5. Qualitative examples of extracting motion concepts from BABEL labels and generating question-answer pairs for BABEL-QA. Motion segments in gray boxes are annotated with the *transition* action in BABEL.

A.4. HumanMotionQA and BABEL-QA

Our HumanMotionQA task and BABEL-QA dataset differ from existing video question-answering datasets in two key ways. First, while existing video QA datasets cover reasoning with actions, we aim to address a more fine-grained human behavior understanding problem (for example, what body part is involved in each action). Second, our dataset lies in a different domain of skeleton-based human motion instead of third-person view videos. Such datasets that consist of skeleton-based human motion and corresponding diverse, natural language question-answer pairs do not previously exist.

The benefits of using skeleton representation are as follows. First, as discussed in previous work on skeleton-based action recognition and localization (Xu et al., 2022; Sun et al., 2022), skeleton-based representation eliminates the nuisances of 2D videos such as lighting changes, background variations, etc, and the 3D joint representation is a more compact human-centric representation. Second, skeleton-based representation can be applied in various applications where videos are not convenient to capture. For example, our skeleton-based neuro-symbolic framework can be generalized to analyze 3D human motion reconstructed from different modalities, for example, motion reconstructed from sparse IMU sensors (Jiang et al., 2022) or egocentric videos (Luo et al., 2021), which enable applications in analyzing everyday activities of people or monitoring actions of physical impaired people (where motions are usually reconstructed from egocentric signal).

In addition, although we categorize our questions into three types, our dataset provides coverage across a variety of aspects of human motion. First, the questions within each question type are diverse. Within each question type, there are numerous motion concepts that can be filtered for, and temporal relations add an additional element of complexity and variation.

Second, the motion sequences have a large variation in terms of types of movements, lengths of sequences, duration of actions, and compositions of different movements. With that said, there is a significant amount of questions pertaining to querying actions, as it is a key temporal feature in motion sequences. We built BABEL-QA from the original real-world dataset, where annotators were asked to write descriptions of the motion sequences, which includes naming all the actions in the video.

A.5. Labeling process

The BABEL-QA labels for frame-level texts and action categories are provided by the BABEL dataset. They were originally collected by showing videos of motion sequences from AMASS to human annotators. The human annotators described a list of actions performed in the motion sequences and delineated start and end times from each of the described actions. From these raw frame-level texts, the authors clustered the labels to map them to a set list of action categories. More information about this process can be found in section 3.4 of the BABEL paper.

In our work, we extract motion concepts by parsing these frame-level label texts and action categories. For actions, we extract non-ambiguous action categories. For body parts and direction, we search through the label texts and extract concepts that are written in the texts. As an example, given the action category / label text pairs of (punch, “punching use right hand”), (kick / foot movements, “air kicking with right foot”), (turn , “turn right”), and (walk, “walk”), from the first segment we can extract *punch* and *right hand* concepts, from the second segment we can extract *kick* and *right foot* concepts, from the third segment we can extract the *right* concept, and from the fourth segment we can extract the *walk* concept. Figure 5 contains qualitative examples of the data creation process.

A.6. Baseline information

We visualize the differences between the MotionCLIP-MLP and MotionCLIP-RNN approaches in Figure 6.

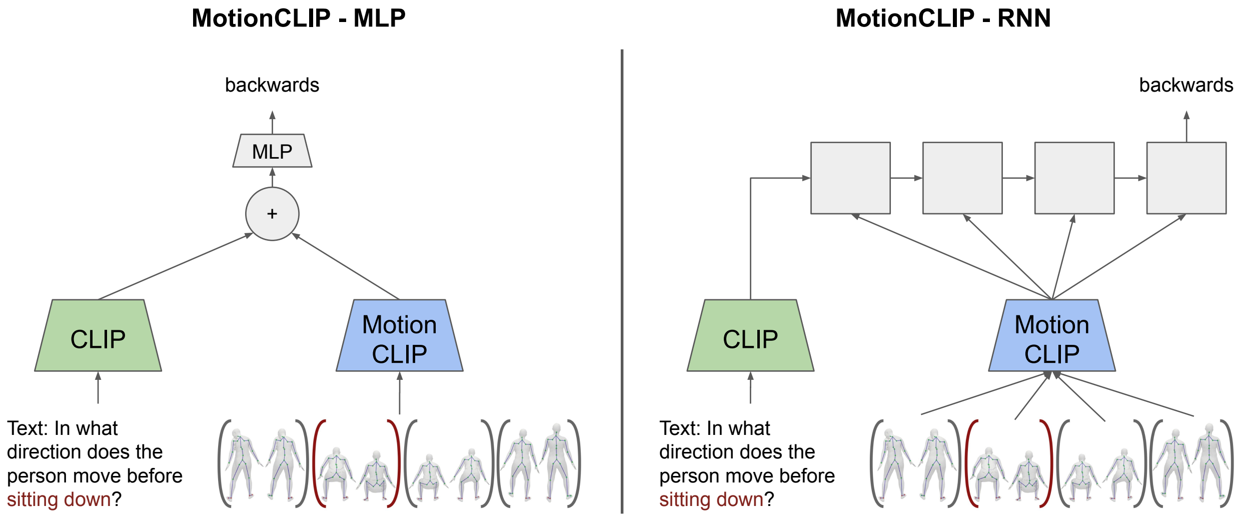

MotionCLIP - MLP
backwards
MLP
+
CLIP
Motion CLIP
Text: In what direction does the person move before sitting down?
MotionCLIP - RNN
backwards
CLIP
Motion CLIP
Text: In what direction does the person move before sitting down?

Figure 6. Visualizations of the MotionCLIP-MLP (left side) and MotionCLUP-RNN (right side) baseline methods.