Supplementary Material for BABEL: Bodies, Action and Behavior with English Labels

Abhinanda R. Punnakkal[*,1] **Arjun Chandrasekaran**[*,1] **Nikos Athanasiou**[1]
Alejandra Quirós-Ramírez[2] **Michael J. Black**[1]
[1]Max Planck Institute for Intelligent Systems, Tübingen, Germany [2]Universität Konstanz, Konstanz, Germany
{apunnakkal, achandrasekaran, nathanasiou, alejandra.quiros, black}@tue.mpg.de

1. BABEL annotation interfaces

Recall that in BABEL, we label the actions in movement sequences in two stages. First, we acquire labels at a sequence level where a single action describes the movement in the entire sequence. Then, if there are multiple actions in the sequence, we collect dense frame-level labels (as discussed in Sec. 3 in the main paper).

In both tasks, we show annotators videos that are rendered from mocap sequences. We ensure that the human figure in the rendered video faces the viewer (camera) in the first frame of the sequence.

Sequence labels. The web annotation interface for sequence-level action labels shows a rendered video of the motion-capture (mocap) sequence. We first ask the annotator the following question – 'Does the video contain more than one action?'. In response, the annotator can choose either 'yes' or 'no'. If the response is 'yes', then we ask the following question – 'If you had to describe the whole video as one action, what would it be?'. On the other hand, if the response is 'no', we instruct the annotator to, 'Name the action:'. In both cases (i.e., when the sequence contains one or multiple actions), we collect the action label that describes the entire sequence – the sequence-level label, via a text-box (free-form labels). The text-box also has an auto-complete feature which matches the person's current input with a fixed list of (typically, single-word) action verbs. The annotator may choose one of the existing actions, or type in a new action. We observe that annotators often enter novel action labels.

We provide text instructions for the task and also provide annotators with example annotations in the interface. A demo of the sequence label annotation interface is available at our project webpage: https://babel.is.tue.mpg.de/demos.html.

Frame labels. We collect frame-level labels for a sequence if the annotators agree that it contains more than one action. We modify the VIA annotation software [3] to suit our requirements. Specifically, we use the video annotation application that allows annotation of temporal segment and spatial regions. We remove the spatial annotation features and customize it for marking 'actions' in the video. We also include tests like making sure there is no gap in the annotation, there is more than one action in the video, etc.

In the frame-level labeling task, the person first watches a video of the movement sequence. Then, they list all the actions occurring the sequence. This includes actions that occur sequentially and simultaneously. Once the annotator has confirmed entering all the actions in a text-box, a horizontal 'timeline bar' is created below the video for each action. The annotator can denote the start and the end time of the action in the sequence, by creating an 'action segment'. To create an action segment, the annotator highlights the timeline for the action of interest, and presses the 'a' key on the keyboard. The start and end times are modified by mouse click and drag operations (or keyboard shortcuts) to accurately reflect the corresponding duration of the action in the video.

Since there is significant variance in the length of the sequences and the number of actions in a sequence, in addition to the fixed pay, we also provide annotators with an optional bonus payment. The bonus payment is proportional to the number of action segments that are labeled per sequence. To encourage annotators to be thorough, we mention this in the instructions.

When multiple actions occur in a sequence, there is often a transition between them. To reduce user effort, and encourage people to explicitly annotate transitions between actions, we populate the list of actions with a 'transition' action by default.

We provide detailed instructions for the task in the interface. In addition, we also ask annotators to watch a video tutorial explaining the interface and task with examples. Recall that, to ensure the quality of annotations, we first pro-

[*] Denotes equal contribution.

vide annotators with a test task and only qualify annotators who demonstrate a clear understanding of our task.

A demo of the frame label annotation interface is available at our project webpage: https://babel.is.tue.mpg.de/demos.html. In the demo, we provide the interface with 2 mocap sequences. The first sequence, a labeled sample from BABEL, illustrates the level of detailed annotation of a completed frame-level labeling task. We leave the second sequence unlabeled in case there is interest in attempting the annotation task.

2. Visualization of samples from BABEL

We provide a search interface to visualize samples from BABEL here: https://babel.is.tue.mpg.de/explore.html. We superimpose the actions corresponding to each frame on top of the video. This makes reviewing the (potentially multiple) action labels for each frame of the sequence easier. Note that the playback speed of the video can be adjusted by clicking on the +/- buttons (second column). For reference, we also list all unique actions in a sequence (under the 'All actions' column).

The densely labeled samples from BABEL illustrate that natural human movement consists of multiple actions that occur simultaneously and sequentially. We observe that even short sequences ($<$ 3 seconds) of natural movement consist of multiple actions. We also observe from these qualitative samples that the transitions between actions are explicitly annotated. Indeed, 'transition' is the most frequently occurring label in BABEL, and the duration of transitions is second-highest among all actions (see Sec. 3 for more details). BABEL uniquely captures detailed information regarding actions that comprise human movement. We believe that it will provide useful data for learning and evaluating statistical models that correlate movement with semantic labels.

3. BABEL labels

Label organization. We provide the action categories and raw action labels in BABEL in https://babel.is.tue.mpg.de/explore.html. Recall that the list of labels in one row denote the member elements of the action category cluster, described in Sec. 3.5 in the main paper.

A visualization of the label organization in the entire BABEL dataset is provided in: https://babel.is.tue.mpg.de/explore.html. Note that the visualization is interactive – hovering the mouse over each semantic category shows, in the 'value' field, the number of action categories within the semantic category. The width of each action category is proportional to the number of raw labels associated with the category, which is also visible under the 'value' field on hovering over each action category.

The label organization plot conveys the hierarchy and diversity of labels for action categories. Note that the area of the sectors in the chart do not correspond to the number of samples or duration of samples from each category. These are discussed in the next section.

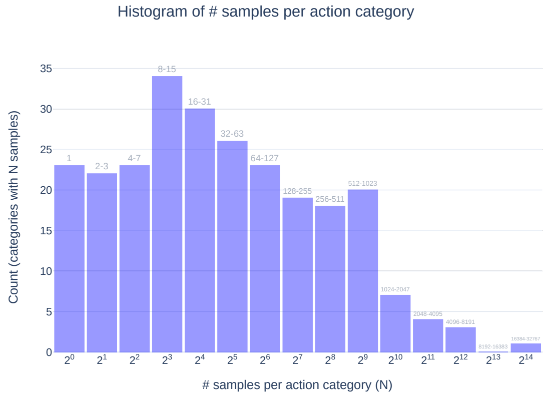

Figure 1. The distribution of the number of samples across action categories in BABEL is long-tailed. Y-axis denotes the number of action categories that contain N samples. X-axis denotes the number of samples belonging to an action category (N) in log scale.

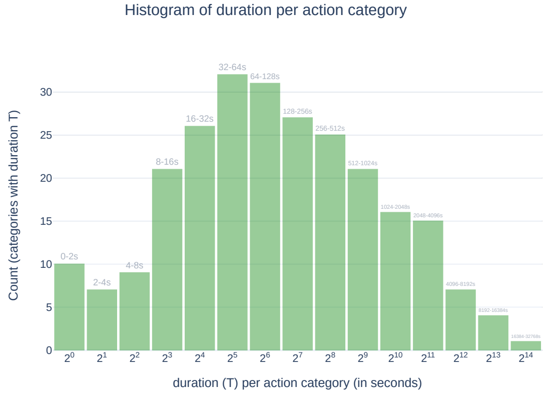

Figure 2. The distribution of the duration across action categories in BABEL is long-tailed but less skewed than the number of samples per category. Y-axis denotes the number of action categories that account for duration T. X-axis denotes the total duration (T) that an action category accounts for, in log scale.

Label distribution. The mocap sequences in BABEL are acquired from AMASS [5] which contains many mocap datasets, as described in Sec. 3.1 of the main paper. Thus, BABEL does not have a strictly controlled distribution of categories, unlike many other datasets. The distribution of action categories in BABEL, is long-tailed, as shown in Fig. 1. Similarly, the overall duration that action categories account for, also follow a long-tailed distribution, as shown in Fig. 2. While we expect that the shape of these

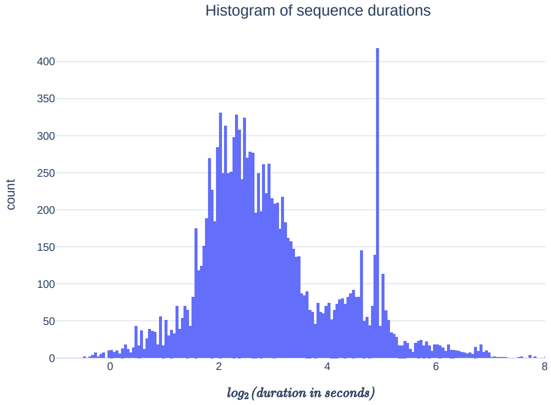

Figure 3. The distribution of the sequence durations in AMASS. Y-axis denotes the number of segments within duration specified by the bin. X-axis denotes durations of bins in `log` scale.

distributions will change as BABEL grows, we do expect the distributions to remain skewed, similar to many naturally occurring distributions. We believe that learning from skewed distributions that occur naturally, is an important, and challenging problem. BABEL has the potential to serve as a benchmark which encourages and reflects progress in the ability to deploy algorithms to real-world applications that have skewed class distributions.

In BABEL, action categories such as 'walk', 'transition', 'stand', etc. occur quite frequently in the dataset, and the frequency of other classes decreases exponentially, following Zipf's law. We visualize the overall number of segments for each action category in BABEL, in `https://babel.is.tue.mpg.de/stats.html`.

4. Potential sources of Bias

4.1. Annotators

BABEL consists of English language action labels for human movements. As such, an important goal during data collection is to recruit annotators who are fluent in English. Further, cultural differences also introduces variability in language. With BABEL, our goal is to keep this variance to a minimum. Thus, to satisfy these two goals, we only recruit annotators from either the US or Canada, via the crowdsourcing platform Amazon Mechanical Turk. As a consequence, the action labels in BABEL may be biased towards English words and phrases prevalent in North-America. In a future study, it would be interesting to study the cross-cultural similarities and differences in perception of movements and action labels.

4.2. AMASS

The mocap sequences in BABEL are derived from AMASS [5], which is a large corpus of mocap datasets. As a consequence, BABEL could have inherited the same biases present in these existing mocap datasets.

Each dataset in AMASS differs in the number and distribution of actions, number of subjects, the duration of the sequence, etc. We examine a few of these in more detail below.

Sequence duration. In Fig. 3, we present a histogram of sequence durations of AMASS mocap sequences. We see that AMASS has a bimodal distribution over sequence durations. Interestingly, we observe a large, narrow spike in the bin of range $(4.90, 4.95)$, indicating the presence of 418 sequences with duration sec., i.e., $(29.86, 30.91)$ sec. $318/418$ sequences in this duration-range belong to the Eyes Japan dataset [1] which are exactly 30.0 sec. long. We note that in real-world, it is likely that we encounter action lengths of varying durations. Training models on carefully constructed movement sequences of fixed constant durations might negatively impact the generalization of the models to real-world data.

Action diversity across datasets. AMASS has significant diversity in the types of actions performed by mocap actors across datasets. For instance, some datasets such as BMLmovi [4] focus on everyday actions, e.g., `walking`, `jogging`, `pretending to talk on the phone`, etc., which are easily recognizable by a lay person. On the other hand, the SFU Motion Capture Database [2] contains movements performed by experts in a fine-grained dance categories, e.g., `cha cha cha`, `xinjiang chinese dance`, and martial arts categories like `Kendo`, `Wushu (Chinese Kungfu)`, etc. Indeed, while a lay person might be able to identify these movements `dance` and `martial arts` actions, domain experts in these areas might provide more specific labels.

For details regarding the individual datasets in AMASS, we refer readers to Mahmood et al. [5], and the relevant papers and websites describing the source datasets.

Actions in the beginning and the end. Real-world data is likely to contain long, continuous streams of movement. Current 3D data however, are of relatively shorter length, and are collected from actors performing actions that are typically 'interesting'. Thus, the data in current datasets is likely qualitatively different from real-world data. To evaluate this in BABEL sequences, we examine the distribution of actions within each sequence.

Given a sequence, we compute the frequency of the actions occurring in the beginning and the end, normalized

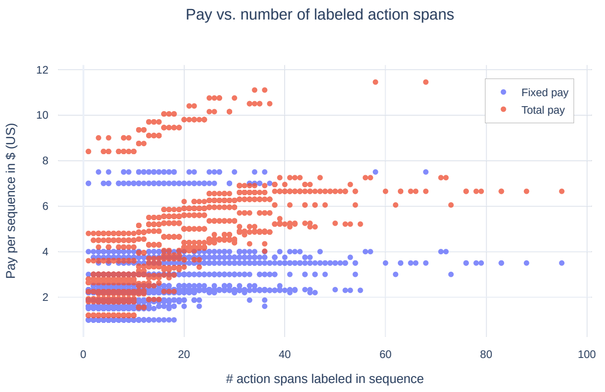

Figure 4. Payment to an annotator in frame labeling task vs. the number of actions they labeled. X-axis denotes the pay in US Dollars for a sequence. Y-axis denotes the number of actions labeled by the annotator in a sequence. We provide a bonus payment in addition to the fixed pay, to ensure that annotators are compensated for their thorough work.

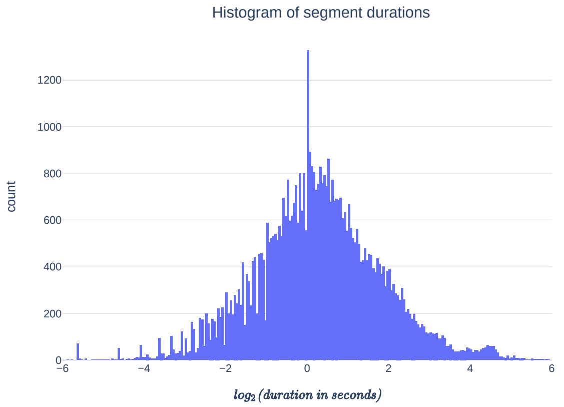

Figure 5. The distribution of the segment durations in BABEL, with a mode around 1 sec. duration. Y-axis denotes the number of segments within duration specified by the bin. X-axis denotes durations of bins in log scale.

by their overall frequency across the dataset. We define 'beginning' and 'end' as the first and last segment of a sequence. According to this normalized frequency, we find that a pose, t pose and stand are the most common actions at the start, which occur 49%, 47%, 35% of their total occurrences in the beginning of a sequence. Overall, they account for the beginning of 3918 (29.6%) sequences in BABEL. Similarly, the actions categories a pose, t pose, stand are the most frequent actions that conclude a sequence, appearing 43%, 42% and 35% of their total occurrences in the end of a sequence. They account for 3744 sequences in BABEL (28.32%). Thus, the distribution of actions at the beginning and end of mocap sequences show distinct biases that are a consequence of calibration.

4.3. Pay

In our initial data collection experiments, we observe that there exists a weak positive correlation between the duration of a sequence and the number of labeled actions. Since the frame labeling task involves naming the action, and precisely marking the span of the action, we assume that the duration of the task will be weakly correlated with overall duration. Thus, in our frame labeling task, we set the pay as a function of the sequence duration.

While the sequence duration is a weak indicator of the number of actions being performed, it is by no means perfect – there exist many relatively short sequences containing many action segments. Since annotators labeling these sequences end up spending more time than we estimated, we introduce an additional bonus payment which is a function of the number of labeled action segments (see Fig. 4). We see that while the fixed pay (blue points) is different for different sequences, they do not appear to be correlated with the number of action segments. With the addition of the bonus payment (red points) however, the overall pay is now better correlated with the number of action segments that the annotator labels.

In addition to providing fair compensation, the bonus also acts as an incentive for annotators to perform a thorough job. While it is true that technically, an annotator could be biased to (incorrectly) label more segments solely motivated by the bonus, the diminishing returns discourages this.

4.4. Action segments

Accurately labeling a movement sequence with all actions and their precise spans is a difficult, attention-demanding task. When an annotator chooses the option to mark the span of an action, a segment is added to the selected action, beginning from the current time of the video. It is likely that the annotator's task will be easier if the span of the default segment is close to the actual span of the action in the video. In our interface, the default segment duration is fixed at 1 sec.

We provide the histogram of the durations of segments in BABEL in Fig. 5. We observe that the distribution of segment durations approximates a log-normal distribution, with the mode of the distribution around 2^0, i.e., 1 sec. However, we also observe a spike around the mode (1 sec.). There are about twice as many segments in the $\sim$ 1 sec. bin, compared to its neighbors. This indicates a potential bias towards 1 sec. segments. We think that a likely reason for this bias could be that the default segment duration in our interface is 1 sec. A possible hypothesis is that when the span of an action in the video is close to 1 sec., anno-

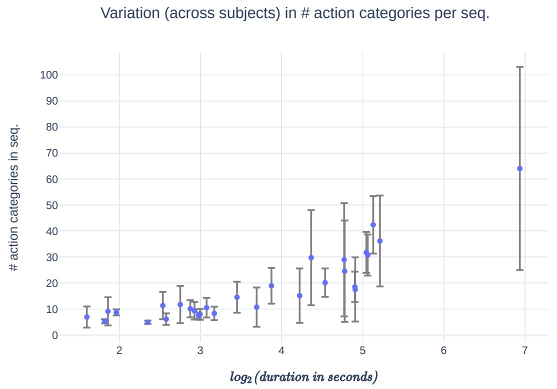

Figure 6. Mean and std. dev. of the number of labeled actions for a sequence, computed across annotators. X-axis denotes the duration (of a sequence) in log scale. Y-axis denotes the number of action categories. Each blue point represents the mean number of actions across annotators, for a sequence. The error bars depict the std. dev. The number of action categories and the variance across annotators seem to increase with increasing sequence duration.

tators avoid additional precise manual editing of the ending of the default segment.

As part of our data quality control process, we visualize many random sequences along with their frame-level levels, and manually verify that the labeling is accurate (with a small error margin). Further, we only make the task available to around 130 annotators whose work we determine to be reliable. So we do not expect the effect of this bias to largely impact the quality of the dataset. Indeed, this bias was not obvious to us from the label visualization — this is apparent only when viewing the histogram of segment durations with a small bin size.

A possible solution for the future could be an interface design where every segment is initialized with a random duration. It would be interesting to: (1) identify if there are still a relatively larger number of 1 sec. segments, and (2) measure how often annotators edit the default segment duration. However, there exists an important trade-off with a design where the default segment duration may be quite different from the duration of majority of the action spans. Having to perform larger, and more frequent edits of segments could negatively impact the overall labeling quality, throughput and annotator satisfaction with the task.

5. Inter-annotator variation

The task of providing semantic descriptions of human actions is often ambiguous and subjective. We investigate the variance in the labeling task by comparing annotations on the same sequence, that are labeled by different, unique annotators. For this analysis, we select sequences containing multiple, diverse actions belonging to different semantic classes. We also ensure that the sequences vary widely in their durations, unlike the $\sim$ log-normal distribution of sequence durations in BABEL shown in Fig. 3. Overall, we experiment with 29 mocap sequences, each labeled by 5 distinct annotators.

Number of action categories. In BABEL, two annotators agree that a sequence contains a single action, about 22.1% of the time (recall the sequence labels presented in Sec. 3.1 in the main paper). The mean number of action categories per sequence across the entire dataset $= 6.68$. Note that we study inter-annotator disagreements on a selected subset that has a mean number of action categories $= 18.06$, which is about 3 times larger.

We first evaluate the inter-annotator variation in the number of action categories per sequence. As shown in Fig. 6, there is a non-trivial variation in the number of actions labeled per sequence. There appears to be a weak correlation between the duration of the sequence, and both the mean and the variance in the number of action categories computed across annotators.

We compute the Coefficient of Variation ($\texttt{CV}$) for each sequence, given the distribution over the number of action categories across annotators. $\texttt{CV}= \sigma/\mu$ (std. dev / mean) measures the dispersion of a frequency distribution. Across 29 sequences, the mean ± std. of $\texttt{CV}= 0.42 \pm 0.20$.

The relatively large dispersion illustrates the inherent variation in the types of human movements in AMASS, and the difficulty in describing the movements with action labels that are nonsubjective. Our project webpage contains examples of challenging sequences. We observe that the person in such sequences, moves freely and performs multiple actions, which are ambiguous to precisely describe. These sequences highlight the challenging nature of our task – precisely aligning continuous movement of humans, to semantically meaningful (action) language labels.

6. BABEL action recognition

Details regarding the skeleton. The mocap sequences in BABEL are derived from the AMASS [5] dataset, which represents the human using SMPL+H [6], a parametric body model. To be consistent with prior work [7, 8], we also use the NTU RGB+D skeleton in our experiments.

Concretely, we map 21/25 joints in the NTU RGB+D skeleton to the corresponding joints in the SMPL+H skeleton based on the joint names. However, the joints named, `tip of {left/right} hand` and `{left/right} thumb` in the NTU RGB+D skeleton are not represented directly in the SMPL+H skeleton. We identify the vertices in the SMPL+H mesh that correspond to these joints in the NTU RGB+D skeleton. This provides us with a mapping that allows us to convert data in the SMPL+H representation to the NTU RGB+D skeleton representation.

Finally, since AMASS only provides pose parameters for SMPL+H in terms of joint rotations, we extract the joint positions (corresponding to the NTU RGB+D skeleton), for the AMASS sequence using the `smplx`[1] python package. We assume a constant mean body shape for all sequences. We visualize the transformed data to verify its validity.

Dataset splits. We benchmark 2 versions of BABEL for the action recognition task, denoted as BABEL-60, and BABEL-120. They contain sequences from BABEL that contain the 60 and 120 most frequent action categories, respectively, with a few exceptions. For instance, labels in BABEL also includes the intermediate frames where a person transitions from one action to another. Since the action being performed in these 'transition' frames are ambiguous, we exclude the 'transition' action label and the corresponding movement segments from the BABEL action recognition splits. The BABEL-60 split corresponds to the first 60 classes of BABEL-120.

The BABEL dataset splits, baseline methods, and evaluation code will be made available and supported for academic research purposes at https://babel.is.tue.mpg.de/.

References

[1] Eyes Japan. (Date last accessed 29-March-2021). 3

[2] SFU Motion Capture Database. (Date last accessed 29-March-2021). 3

[3] Abhishek Dutta and Andrew Zisserman. The VIA annotation software for images, audio and video. In *Proceedings of the 27th ACM International Conference on Multimedia, MM 2019, Nice, France, October 21-25, 2019*, pages 2276–2279. ACM, 2019. 1

[4] Saeed Ghorbani, Kimia Mahdaviani, Anne Thaler, Konrad Kording, Douglas James Cook, Gunnar Blohm, and Nikolaus F. Troje. Movi: A large multipurpose motion and video dataset, 2020. 3

[5] Naureen Mahmood, Nima Ghorbani, Nikolaus F. Troje, Gerard Pons-Moll, and Michael J. Black. AMASS: Archive of motion capture as surface shapes. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 5441–5450. IEEE, 2019. 2, 3, 5

[6] Javier Romero, Dimitrios Tzionas, and Michael J. Black. Embodied hands: Modeling and capturing hands and bodies together. *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*, 36(6), Nov. 2017. 5

[7] Amir Shahroudy, Jun Liu, Tian-Tsong Ng, and Gang Wang. NTU RGB+D: A large scale dataset for 3d human activity analysis. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 1010–1019. IEEE Computer Society, 2016. 5

[8] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. Two-stream adaptive graph convolutional networks for skeleton-based action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 12026–12035. Computer Vision Foundation / IEEE, 2019. 5

[1] https://github.com/vchoutas/smplx