## License of the assets

### Licence for the codes

We use the code for MS-TCN [13], ASRF [24], LAS [9], all of which are under MIT License according to `https://opensource.org/licenses/MIT`.

### Licence for the dataset

50salads [52] is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License.
Breakfast [32] by H. Kuehne, A. B. Arslan and T. Serre is licensed under a Creative Commons Attribution 4.0 International License.
For the Jigsaws [18] dataset, we follow the data use agreeement according to `https://cs.jhu.edu/~los/jigsaws/info.php`.

# A State of the art in action sequence identification

**Action classification**: Action classification is the task of identifying a single action, as opposed to a sequence of actions. Several methods use 2D CNNs to extract frame-wise features from an input video, which are then combined to predict a coarse action taking place in the video [56, 39, 59]. Alternatively, techniques such as C3D [54], I3D [8] SlowFast [15] and X3D [14] use 3D CNNs to exploit the spatial-temporal information in the data. There also exist several works that perform action classification from kinematic data [2, 12].
**Action segmentation**: Action segmentation is the problem of segmenting an input stream of data, labeling each frame according to the action that is being carried out. Earlier methods for action segmentation employed hidden Markov models [33, 22]. More recently, convolutional neural networks [58, 26] and recurrent neural networks [50] have been applied to this problem Inspired by the success of temporal convolutional networks (TCNs) in speech synthesis, [37] adapted these models to action segmentation. MS-TCN [13], which uses a multi-stage TCN architecture, has become one of the most widely used architecture for action segmentation. Although these methods achieve high frame-wise accuracy, they still produce a significant number of over-segmentation errors. In order to address this, several boundary-aware methods have been developed which perform temporal smoothing of the frame-wise predictions [57, 24]. These methods use ground-truth boundary information to train a binary classification network to perform boundary detection. The boundary estimates are then used to aggregate the frame-wise predictions either in a soft manner (boundary-aware pooling) or by setting a hard threshold. However, for elemental actions with a short duration, such as the functional primitives in the StrokeRehab dataset, the duration of each action is very short. As a result, the boundaries between actions can be hard to detect or even hard to define (see Figure 4).
**Sequence-to-sequence models**: Our proposed method is based on sequence-to-sequence (seq2seq) models. These models allow us to learn a mapping of a variable-length input sequence to a variable-length output sequence [53]. Popular sequence-to-sequence models include the RNN-transducer [20], connectionist temporal classification (CTC) based models [21] and encoder-decoder based models [9]. Here we leverage encoder-decoder based models, which have very successful in machine translation [3, 40], speech recognition [9] and image captioning [55]. There are also some recent work to apply the seq2seq model for action segmentation under weak supervision [51], action forecasting [44].

# B Additional results for comparison of segmentation-based models and seq2seq models

## B.1 Trained only on healthy subjects

Table 4 shows the performance of segmentation-based models and seq2seq models when trained only on the healthy subjects and evaluated on both the healthy and stroke-impaired test set.

## B.2 Trained only healthy subjects and stroke-impaired patients

Table 5 shows the performance of segmentation-based models and seq2seq models when trained only on the healthy subjects and the stroke-impaired patients; and evaluated on both the healthy and stroke-impaired test set. We do see slight improvement in the performance for healthy subjects

Table 4: Results on *StrokeRehab* Healthy controls dataset. We trained and evaluated the model on the train and test set respectively which consists the healthy subjects. We report mean (95% confidence interval) which is computed via bootstrapping (see Appendix G.2). * indicates models selected based on the best validation frame-wise accuracy

| | Model | Healthy subjects test set | | Stroke test set | |
|---|---|---|---|---|---|
| | | Edit Score | Action Error Rate | Edit Score | Action Error Rate |
| Segmentation-based model | MS-TCN* [13] | 69.85 (68.22 - 71.47) | 0.333 (0.309 - 0.358) | 62.47 (61.00 - 63.85) | 0.434 (0.409 - 0.463) |
| | MS-TCN [13] | 70.56 (68.51 - 72.15) | 0.337 (0.314 - 0.367) | 62.19 (60.94 - 63.53) | 0.467 (0.438 - 0.501) |
| | + Smoothing window | 72.21 (70.65 - 73.61) | 0.291 (0.272 - 0.311) | 63.70 (62.21 - 65.27) | 0.397 (0.377 - 0.417) |
| | ASRF* [24] | 71.13 (69.58 - 72.70) | 0.320 (0.299 - 0.341) | 62.06 (60.67 - 63.64) | 0.436 (0.410 - 0.460) |
| | ASRF [24] | 70.79 (69.09 - 72.39) | 0.329 (0.305 - 0.356) | 62.14 (60.66 - 63.47) | 0.463 (0.437 - 0.492) |
| Seq2seq | Seg2seq | 70.97 (69.57 - 72.41) | 0.296 (0.279 - 0.316) | 62.24 (60.67 - 63.78) | 0.412 (0.391 - 0.438) |
| | Raw2seq | 72.44 (71.06 - 73.88) | 0.281 (0.263 - 0.299) | 61.10 (59.54 - 62.54) | 0.405 (0.383 - 0.425) |

Table 5: Results on *StrokeRehab* Healthy controls and stroke impaired patients. We trained and evaluated the model on the train and test set respectively which consists the healthy subjects and mildly + moderately impaired patients. We report mean (95% confidence interval) which is computed via bootstrapping (see Appendix G.2). * indicates models selected based on the best validation frame-wise accuracy

| | Model | Healthy subjects test set | | Stroke test set | |
|---|---|---|---|---|---|
| | | Edit Score | Action Error Rate | Edit Score | Action Error Rate |
| Segmentation-based model | MS-TCN* [13] | 70.77 (68.80 - 72.41) | 0.322 (0.296 - 0.350) | 66.76 (65.25 - 68.56) | 0.380 (0.349 - 0.411) |
| | MS-TCN [13] | 70.76 (69.19 - 72.38) | 0.306 (0.284 - 0.329) | 67.42 (66.02 - 69.01) | 0.357 (0.334 - 0.377) |
| | + Smoothing window | 71.40 (69.54 - 73.16) | 0.288 (0.267 - 0.307) | 67.54 (66.00 - 69.21) | 0.331 (0.309 - 0.353) |
| | ASRF* [24] | 71.72 (69.98 - 73.51) | 0.305 (0.283 - 0.328) | 67.39 (65.85 - 68.90) | 0.345 (0.324 - 0.367) |
| | ASRF [24] | 70.98 (69.26 - 72.49) | 0.307 (0.287 - 0.332) | 67.51 (66.08 - 68.94) | 0.357 (0.337 - 0.378) |
| Seq2seq | Seg2seq | 67.47 (65.75 - 69.18) | 0.327 (0.307 - 0.349) | 64.50 (62.70 - 66.16) | 0.345 (0.321 - 0.369) |
| | Raw2seq | 72.06 (70.62 - 73.64) | 0.287 (0.269 - 0.304) | 69.35 (67.83 - 70.90) | 0.297 (0.279 - 0.318) |

and stroke-impaired patients when the model is trained on the combined data from healthy and stroke-impaired subjects.

# C Additional result for distribution shift in StrokeRehab dataset

Table 3 in the main paper showed the results of model generalization when the raw2seq model is trained and tested on various combination of subject cohorts. Table 6 shows the same result but reports the Edit score instead of the Action Error Rate (AER)

# D False Discovery Rate and True Positive rate

In addition to edit score and action error rate, we also evaluated the model performance using the true positive rate (TPR) and the false discovery rate (FDR). To compute these quantities we compared the estimated sequence to the ground truth actions to determine the number of actions that are correctly identified, incorrectly identified (e.g. the model estimates a transport, but the ground truth is a reach), missed (an action present in the ground truth is not present in the estimate), or spurious (an action present in the estimate is not present in the ground truth). The TPR is the ratio between the correctly-identified actions and the total ground-truth actions. The FDR is the ratio between the wrong predictions (incorrectly identified and spurious) and the total estimated actions. It should be noted that there is a trade-off between TPR and FDR. Therefore, to compare the various models, both TPR and FDR can be combined using F1-score, which is the harmonic mean of TPR and 1-FDR: $\mathrm{F1} := \frac{2(1-\mathrm{FDR})\,\mathrm{TPR}}{1-\mathrm{FDR}+\mathrm{TPR}}$.

# E Additional analysis for study on distribution shift

In Table 3, we show that the model trained only on healthy subjects failed to generalize to the other two cohorts (i.e. stroke patients and severely impaired patients). In contrast, models trained on stroke patients generalize well to healthy subjects. We provide additional analysis as follows.

The general structure of the functional movements is similar in healthy subjects and stroke patients, although it tends to be more regular in healthy patients. Therefore the model trained on stroke patients is able to learn "normal" functional movements that enable it to perform well on healthy subjects. The reason why it does not perform as well on the test stroke patients is that the movements of

Table 6: In order to study the effect of distribution shift on StrokeRehab we evaluate a Raw2seq model trained only on healthy subjects (HS), only on stroke patients (SP) and on both (HS+SP) using different test datasets (see Section 2.4). The model trained only on healthy subjects, fails to generalize to the other two cohorts. In contrast, models trained on stroke patients generalize well to healthy subjects. All models have difficulties generalizing to severely impaired patients (but the performance of the HS-trained model is particularly poor). We report the mean of the metrics of interest with 95% confidence intervals computed via bootstrapping (see Appendix G.2).

| Tested on | Healthy subjects | Stroke patients | Severely impaired |
|---|---|---|---|
| Trained on | Edit Score | Edit Score | Edit Score |
| Healthy Subjects (HS) | 72.4 (71.0 - 73.8) | 61.1 (59.5 - 62.5) | 35.3 (32.7 - 38.1) |
| Stroke patients (SP) | 72.6 (70.9 - 74.0) | 68.8 (67.4 - 70.3) | 46.1 (43.0 - 48.8) |
| HS + SP | 72.0 (70.6 - 73.6) | 69.3 (67.8 - 70.9) | 49.0 (45.9 - 52.2) |

Table 7: Comparison of seq2seq model and action segmentation model on *StrokeRehab*: in terms of FDR and TPR.

| | Model | Video Data | | Sensor Data | |
|---|---|---|---|---|---|
| | | FDR | TPR | FDR | TPR |
| Segmentation-based model | MS-TCN* [13] | 0.130 (0.118 - 0.143) | 0.621 (0.600 - 0.642) | 0.229 (0.207 - 0.249) | 0.785 (0.765 - 0.806) |
| | MS-TCN [13] | 0.148 (0.135 - 0.161) | 0.643 (0.622 - 0.664) | 0.201 (0.186 - 0.216) | 0.790 (0.770 - 0.812) |
| | + Smoothing window | 0.180 (0.166 - 0.194) | 0.666 (0.646 - 0.685) | 0.162 (0.149 - 0.173) | 0.758 (0.737 - 0.779) |
| | ASRF* [24] | 0.114 (0.102 - 0.126) | 0.564 (0.542 - 0.587) | 0.172 (0.156 - 0.185) | 0.747 (0.727 - 0.769) |
| | ASRF [24] | 0.126 (0.113 - 0.139) | 0.585 (0.566 - 0.605) | 0.219 (0.205 - 0.233) | 0.784 (0.763 - 0.803) |
| Seq2seq | Seg2seq | 0.211 (0.200 - 0.221) | 0.743 (0.732 - 0.753) | 0.145 (0.134 - 0.157) | 0.707 (0.678 - 0.734) |
| | Raw2seq | 0.216 (0.207 - 0.226) | 0.734 (0.722 - 0.744) | 0.166 (0.153 - 0.179) | 0.767 (0.747 - 0.786) |

stroke patients tend to be very heterogeneous, which makes action recognition more challenging than for healthy subjects (especially for the test patients with higher impairment). We observe the same phenomenon for the segmentation-based models (see Table 5).

# F Confusion matrices with insertion and deletions values

Table 9 shows the confusion matrix in Table 7 with the values of insertions and deletions for the IMU data. Similarly, Table 10 shows the confusion matrix in Table 7 with the values of insertions and deletions for the video data.

# G Experimental details

## G.1 Benchmark datasets

**50 salads dataset**: 50 salads dataset [52] contains 50 videos with 17 action classes. In this dataset, 25 people in total prepare two kinds of mixed salads. Each video contains 9000 to 19000 RGB frames. On average, each video contains 20 action instances and is 6.4 minutes long. For evaluation, we follow [24], performing five-fold cross-validation and reporting the average. As per [13], we use I3D model [8], pretrained on the Kinetics dataset, to extract spatio-temporal features from the videos and then use them as the input for our models.

**Breakfast dataset**: Breakfast dataset [32] contains 1712 videos, which are recorded in 18 different kitchens displaying activities of breakfast preparation. There are 48 different actions. On average, each video contains 6 action instances. For evaluation, we follow [24], performing four-fold cross-validation and reporting the average. As per [13], we use I3D model [8], pretrained on the Kinetics dataset, to extract spatio-temporal features from the videos and then use them as the input for our models.

**Jigsaws dataset**: Jigsaws dataset [18] contains 103 surgical activities of 3 types - Knot tying, Suturing and Needle Passing performed by 8 subjects using a robotic surgical system. The subjects have varying degrees of robotic surgical experience. The kinematic data from the robotic arm and the videos of the activities are available. The activities contain 14 different actions overall. On average, each video contains 17 action instances. We split the subjects into 4 folds,where each fold contains two subjects with different levels of robotic experience. For evaluation, we follow [24], performing four-fold cross-validation and reporting the average.

Table 8: Comparison of seq2seq model and action segmentation model on 50Salads, Breakfast, Jigsaws: in terms of FDR and TPR.

| | Model | 50 salads | | Breakfast | | Jigsaws | |
|---|---|---|---|---|---|---|---|
| | | FDR | TPR | FDR | TPR | FDR | TPR |
| Segmentation-based model | MS-TCN* [13] | 0.28 | 0.84 | 0.34 | 0.82 | 0.435 | 0.906 |
| | MS-TCN [13] | 0.26 | 0.86 | 0.34 | 0.82 | 0.357 | 0.898 |
| | + Smoothing window | 0.18 | 0.83 | 0.23 | 0.79 | 0.174 | 0.878 |
| | ASRF* [24] | 0.18 | 0.80 | 0.19 | 0.77 | 0.246 | 0.851 |
| | ASRF [24] | 0.17 | 0.80 | 0.19 | 0.77 | 0.191 | 0.846 |
| Seq2seq | Seg2seq | 0.17 | 0.82 | 0.17 | 0.79 | 0.092 | 0.874 |
| | Raw2seq | 0.31 | 0.76 | 0.28 | 0.70 | 0.218 | 0.780 |

Table 9: Confusion matrix with the values of insertions and deletions for IMU data. For each row, the first six entries would add to one. Since, we are comparing a predicted sequence to a ground truth sequence, there are some substitutions, insertions and deletions needs to be done to the predicted sequence so that we reach the ground truth sequence. So, substitutions (first 5 columns) and insertions are the primitives that make up the ground truth sequence. Deletions are spuriously predicted primitives that are not part of the ground truth sequence.

| Confusion matrix with insertions and deletions | | Prediction (Substitutions) | | | | | Insertions | Deletions |
|---|---|---|---|---|---|---|---|---|
| | | Idle | Reach | Reposition | Stabilize | Transport | | |
| Ground truth | Idle | 0.792 | 0.008 | 0.004 | 0.037 | 0.032 | 0.127 | 0.076 |
| | Reach | 0.005 | 0.776 | 0.012 | 0.041 | 0.046 | 0.12 | 0.044 |
| | Reposition | 0.003 | 0.020 | 0.754 | 0.037 | 0.018 | 0.168 | 0.065 |
| | Stabilize | 0.024 | 0.041 | 0.014 | 0.677 | 0.013 | 0.231 | 0.136 |
| | Transport | 0.020 | 0.026 | 0.013 | 0.022 | 0.822 | 0.097 | 0.064 |

### G.2 Validation and Evaluation

In the case of the action segmentation models, we use a small validation set separate from the training set of each fold to select models based on Action Error Rate (AER), and test the performance on the validation set of that fold. We then follow the same evaluation as previous works [13, 24] by simply averaging the result for all the folds. For StrokeRehab, there is a held-out test set. To perform model selection, we apply 4-fold cross-validation. For each fold, we select the best model according to the best Action Error Rate (AER) on the validation set. We then ensemble the prediction of the models trained on 4 folds and evaluate it on the held-out test set. For the action segmentation models, we perform ensembling by averaging the model outputs. In the case of the seq2seq model, the decoding of each action depends on the previous prediction. To ensemble, average the output of the models at each step and use this average prediction as the previous prediction in the decoder for each individual model.

**Confidence interval computation**: We report the 95% confidence interval for the StrokeRehab dataset by creating 1,000 bootstrap replicates of the test set. The upper and lower confidence limit are the 97.5 percentile and 2.5 percentile of the performance measure of interest respectively, computed using the 1,000 bootstrap replicates of the test set.

### G.3 Seq2seq models

**Encoding**: The encoder $f_{\text{enc}}$, which can be a convolutional or recurrent network, maps the input sequence $\mathbf{x}$ to a fixed-length hidden vector $\mathbf{h}(\theta_{\text{enc}}) = f_{\text{enc}}(\mathbf{x}; \theta_{\text{enc}})$ ($\theta_{\text{enc}}$ are the parameters of the encoder). The hidden vector must capture any long-term dependencies in the input sequence, which can be challenging in some cases. For such cases, we have observed that using features from a pre-trained segmentation model can boost performance (see Appendix G.4).

**Decoding**: The decoder is a recurrent neural network (RNN), which outputs the estimated sequence based on the hidden vector $\mathbf{h}$. For $i = 1, 2, \ldots$, the decoder estimates the conditional probability of the next action given the previous action and $\mathbf{h}$. To this end, the RNN $f$ maintains a decoder state $\mathbf{s_i}$, which is updated based on the previous action and the hidden vector:

$$\mathbf{s_i}(\theta_{\text{dec}}) := f_{\text{dec}}(\mathbf{s_{i-1}}, \mathbf{h}, y_{i-1}; \theta_{\text{dec}}), \tag{3}$$

where $\theta_{\text{dec}}$ denotes the parameters of the decoder. For $i = 1$, the previous action is set to a *start-of-sequence* token. The conditional probability of $y_i$ given the previous actions is then approximated

Table 10: Confusion matrix with the values of insertions and deletions for video data. For each row, the first six entries would add to one. Since, we are comparing a predicted sequence to a ground truth sequence, there are some substitutions, insertions and deletions needs to be done to the predicted sequence so that we reach the ground truth sequence. So, substitutions (first 5 columns) and insertions are the primitives that make up the ground truth sequence. Deletions are spuriously predicted primitives that are not part of the ground truth sequence.

| Confusion matrix with insertions and deletions | | Prediction (Substitutions) | | | | | Insertions | Deletions |
|---|---|---|---|---|---|---|---|---|
| | | Idle | Reach | Reposition | Stabilize | Transport | | |
| Ground truth | Idle | 0.680 | 0.049 | 0.021 | 0.057 | 0.008 | 0.185 | 0.044 |
| | Reach | 0.032 | 0.847 | 0.023 | 0.008 | 0.028 | 0.062 | 0.071 |
| | Reposition | 0.015 | 0.022 | 0.734 | 0.033 | 0.004 | 0.192 | 0.088 |
| | Stabilize | 0.040 | 0.007 | 0.025 | 0.750 | 0.035 | 0.143 | 0.104 |
| | Transport | 0.008 | 0.044 | 0.005 | 0.037 | 0.709 | 0.197 | 0.066 |

using a multilayer perceptron (MLP) with a softmax output. Specifically,

$$\mathbf{p}_i(\theta_{\text{dec}}, \theta_{\text{enc}}, \theta_{\text{mlp}}) := \text{Softmax}(\text{MLP}(\mathbf{s_i}, \mathbf{h}; \theta_{\text{mlp}})) \tag{4}$$

is a $c+1$-dimensional vector that contains the estimates of the conditional probability that $y_i$ equals each of the $c$ possible actions or an *end-of-sequence* token. The $i$th predicted action $\hat{y}_i$ is obtained by maximizing this conditional probability. The procedure continues until $\hat{y}_i$ equals the *end-of-sequence* token. Optionally, we can incorporate an attention mechanism during decoding [3, 11] (see Supplementary Material).

**Training**: For the sake of simplicity, we consider a single training example $(\mathbf{x}, \mathbf{y})$. The last entry of $\mathbf{y}$ contains the *end-of-sequence* token. The parameters of the encoder and decoder are learned by maximizing the objective function:

$$\max_{\theta_{\text{enc}}, \theta_{\text{dec}}, \theta_{\text{mlp}}} \sum_i \log(\mathbf{p}_i[y_i]), \tag{5}$$

where we omit the dependence of $\mathbf{p}_i$ on $\theta_{\text{enc}}, \theta_{\text{dec}}, \theta_{\text{mlp}}$ to ease notation. Here $\mathbf{p}_i[y_i]$ denotes the probability that the model assigns to the true observed action $y_i$ when receiving $\mathbf{x}$ as an input.

During inference the ground-truth, $\mathbf{y}$ is not available, so the model can only use the previous predicted action $\hat{y}_{i-1}$ to compute $\mathbf{s_i}$ in (3). This suggests using the estimate also during training, a technique known as curriculum learning [5]. We apply this technique by replacing $y_{i-1}$ with $\hat{y}_{i-1}$ in (3) with a probability $\epsilon$, which is increased gradually during training.

**Inference**: During inference, we perform greedy-decoding to find the most likely sequence of actions given the input data. Specifically, at each time step $i$, we use the previous prediction, $\hat{y}_{i-1}$, in (3) to compute the decoder state, $\mathbf{s_i}$, which in turn is used to compute $\mathbf{p_i}$ using (4). Then, we choose $\arg\max_{1 \le j \le c+1} \mathbf{p_i}[j]$ as the predicted action for step $i$ (note that there are $c+1$ possible actions because one is the *end-of-sequence* token). Beam-search decoding is often preferred over greedy-decoding in speech recognition and natural language applications [9], but here it did not provide a significant improvement.

### G.4 Implementation of models

All experiments were conducted on NVIDIA V100 GPUs.

#### G.4.1 Action Segment Refinement Framework (ASRF)

The ASRF model has two modules: one for frame-wise action segmentation and another for boundary detection. The backbone of both modules is a MS-TCN model with several convolutional stages, each composed of multiple layers of dilated residual convolutions. To implement MS-TCN model, we just use the segmentation module. To implement ASRF model, we refine the output of the segmentation module using the boundaries detected by the boundary detection module.

The loss function used to train this model is combination of two loss functions: weight-cross entropy for frame-wise action classification and boundary detection. We use a regularization parameter $\lambda$ to determine the weight of the boundary-detection term in the loss.

Below we provide some specific implementation details for each specific dataset.

**StrokeRehab**: We use a backbone MS-TCN model with 4 convolutional stages where each stage has 10 layers of dilated residual convolutions, outputting 64 channels. For the sensor data, the parameter

$\lambda$ in the loss function was set to 0.1. For the video data, $\lambda$ was set to 1. These values were determined based on preliminary cross-validation experiments on the training data.

**50Salads and Breakfast**: We follow the settings described in the original paper [24]. In particular, we use the same backbone MS-TCN model as for the StrokeRehab dataset. The only difference is that we perform model selection based on the best validation AER rather than the frame-wise accuracy. We do this to achieve a fair comparison with the seq2seq models, since AER is our metric of interest.

**Jigsaws**: We use a backbone MS-TCN model with 4 convolutional stages where each stage has 15 layers of dilated residual convolutions, outputting 128 channels. The $\lambda$ parameter for the boundary detection loss is set to 0.1, based on preliminary cross-validation experiments on the training data.

### G.4.2 Sequence-to-sequence (Seq2seq)

All seq2seq models are trained with the following hyper-parameters: learning rate = 5e-4, dropout = 0.1, weight decay = 0.0001, num of epochs = 150. We ran preliminary experiments to select the hyper-parameters like the dimensionality of hidden representation for RNNs, number of channels and stages in the MS-TCN, learning rate, dropout rate and weight decay. All the seq2seq models are trained on windowed data. During training, we train with overlapping windows of a specific size which varies across models and datasets. During inference, we concatenate the outputs from non-overlapping windows and remove any duplicates at the border of two windows. When dividing the whole sequence into time windows of equal sizes, we zero pad the last window to make sure all the windows in the dataset are of the same size.For the model that use the attention mechanism, we follow the RNN-based transducer [11, 3] for the decoder.

**StrokeRehab video dataset**:

The encoder module is an MS-TCN model with 4 convolutional stages where each stage has 15 layers of dilated residual convolutions, outputting 256 channels. The decoder module is an LSTM with attention mechanism and a 512-dimensional hidden representation.

- *Raw2seq*. During training, we divide each video sequence of 432 dimensional raw feature vectors into 240-frame windows with an overlap of 100 frames. We then train the model using the windows. During inference, we divide each testing video sequence into non-overlapping 240-frame windows and input that to the model. We then concatenate the resulting estimates from each non-overlapping window and remove the duplicates.
- *Seg2seq*. We use the baseline segmentation model to obtain frame-wise prediction probabilities. We then input the frame-wise probabilities to the seq2seq model. During training, we divide the softmax probabilities of each video sequence into 240-frame windows with an overlap of 100 frames and train the model using these windowed inputs. During inference, we divide each testing video sequence into non-overlapping 240-frame windows and input them to the model. We then concatenate the result from the non-overlapping window and remove the duplicates.

**StrokeRehab sensor dataset**: In order to apply the seq2seq models to the sensor data, we window the input sequence. During training, we use overlapping 600 frames windows, where the labels only correspond to the middle 400 frames. We extract overlapping windows with a stride of 50 frames. During inference, we use non-overlapping windows and concatenate the result, removing duplicates. In addition to the seq2seq cost function, we also incorporate a segmentation frame-wise loss on the output of the encoder.

- *Raw2seq*. The input to the model are 77-dimensional sensor measurements. The encoder module is a three layered bi-GRU model with a 3072 dimensional hidden representation. The decoder module is also an GRU with a 6144 dimensional hidden representation.
- *Seg2seq*. We use the baseline segmentation model to obtain frame-wise prediction probabilities. We then input the frame-wise probabilities to the seq2seq model. The encoder module is a three layered bi-GRU model with 256 dimensional hidden representation. The decoder module is also an GRU with 512 dimensional hidden representation.

**50Salads**: The encoder module is an MS-TCN model with 4 convolutional stages where each stage has 15 layers of dilated residual convolutions, outputting 256 channels. The decoder module is an LSTM with attention mechanism and a 512-dimensional hidden representation.

- *Raw2seq*. During training, we divide each video sequence of 1600 dimensional raw feature vectors to 500-frame windows with overlap of 100 frames. During inference, we divide each

testing video sequence into non-overlapping 1600-frame windows. We then concatenate the result and remove duplicates.

- *Seg2seq*. We use the baseline segmentation model to obtain frame-wise prediction probabilities. We then input the frame-wise probabilities to the seq2seq model. During training, we divide the probabilities of each video sequence into 450-frame windows with an overlap of 100 frames. We then train the model using the windowed data. During inference, we divide each testing video sequence into non-overlapping 450-frame windows. We then concatenate the result and remove duplicates.

**Breakfast**: The encoder module is an MS-TCN model with 4 convolutional stages where each stage has 15 layers of dilated residual convolutions, outputting 256 channels. The decoder module is an LSTM with attention mechanism and a 512-dimensional hidden representation.

- *Raw2seq*. During training, we divide each video sequence of 1600 dimensional raw feature vectors into 500-frame windows with an overlap of 100 frames. During inference, we divide each testing video sequence to non-overlapping 1600-frame windows. We then concatenate the result and remove duplicates.
- *Seg2seq*. We use the baseline segmentation model to obtain frame-wise prediction probabilities. We then input the frame-wise probabilities to the seq2seq model. During training, we divide the probabilities of each video sequence into 800-frame windows with an overlap of 200 frames. During inference, we divide each testing video sequence into non-overlapping 800-frame windows. We then concatenate the result and remove duplicates.

**Jigsaws**:

- *Raw2seq*. Seq2seq for raw features of Jigsaws has an encoder, a decoder module and attention mechanism. The encoder is a MS-TCN model with 4 stages of convolutions with each stage having 10 layers of dilated residual convolutions outputting 128 channels. The decoder is a GRU-RNN with 256 dimensional hidden representation. We also used a multi-headed attention mechanism (2 heads) with a multi-layer perceptron producing 128 dimensional representation. During training, we divide each Kinematic input sequence of 38 dimensional raw feature vectors to 400-frame windows with an overlap of 350 frames. We then train the model using these windows. During inference, we divide each testing video sequence to non-overlapping 400-frame windows and input that to the model. We then concatenate the result from the non-overlapping window and remove the duplicates.
- *Seg2seq*. Seq2seq with action segmentation model has an encoder, a decoder module and attention mechanism. The encoder is a MS-TCN model with 4 stages of convolutions with each stage having 5 layers of dilated residual convolutions outputting 64 channels. The decoder is a LSTM with 256 dimensional hidden representation. We also used a single-headed attention mechanism with a multi-layer perceptron producing 64 dimensional representation. We first train an ASRF segmentation model to obtain raw frame-wise prediction probability without refinement, with the 4 stage MS-TCN architecture (4 stacked single stage TCNs). We treat the frame-wise softmax probabilities as the input to the seq2seq model. Seq2seq model has an encoder module and a decoder module. During training, we input the entire input sequence to model without windowing it. The seq2seq model is trained with a primitive level cross-entropy loss. During inference,we input the entire input sequence to model without windowing it and remove the duplicates.

# H Additional description of StrokeRehab

Figure 8 shows the placement of sensors on a subject's body.

## H.1 Description of the Rehabilitation Activities

Tables 11 and 12 describe the activities performed by the stroke patients.

## H.2 Description of the Joint Angles

As described in Section 2.3, the sensor measurements are used to compute 22 anatomical angle values using a rigid-body skeletal model scaled to the patient's height. Table 13 describes these joint angles in detail.

Figure 8: The location of sensors placed on a subject's body. Sensors are lightweight (34 g) and small (matchbook-size). They are adhered to the back of the hand with thin tape that does not interfere with finger movement or grasp. Similarly, the straps holding the sensors to the forearms and arms do not cross any joints. Neither location nor the methods used to affix the sensors are expected to interfere with natural motion.

Table 11: Description of the activities performed by the stroke impaired patients in the cohort (1/2).

| Activity | Workspace | Target object(s) | Instructions |
|---|---|---|---|
| Washing face | Sink with a small tub (32.3 x 24.1 x 2.5 cm³) in it and two folded washcloths on either side of the countertop, 30 cm from edge closest to patient | Washcloths, faucet handle, and tub | Fill tub with water, dip washcloth on the right side into water, wring it, wiping each side of their face with wet washcloth, place it back on countertop. Use washcloth on the left side to dry face, place it back on countertop |
| Applying deodorant | Tabletop with deodorant placed at midline, 25 cm from edge closest to patient | Deodorant (solid twist-base) | Remove cap, twist base a few times, apply deodorant, replace cap, untwist the base, put deodorant on table |
| Hair combing | Tabletop with comb placed at midline, 25 cm from edge closest to patient | Comb | Pick up comb and comb both sides of head |
| Don/doffing glasses | Tabletop with glasses placed at midline, 25 cm from edge closest to patient | Pair of glasses | Wear glasses, return hands to table, remove glasses and place on table |
| Eating | Table top with a standard-size paper plate (21.6 cm diameter) placed at midline, 2 cm from edge, utensils placed 3 cm from edge, 5 cm from either side of plate, a baggie with a slice of bread placed 25 cm from edge, 23 cm left of midline, and a margarine packet placed 32 cm from edge, 17 cm right of midline | Paper plate, fork, knife, re-sealable sandwich baggie, slice of bread, single-serve margarine container | Remove bread from plastic bag and put it on plate, open margarine pack and spread it on bread, cut bread into four pieces, cut off and eat a small bite-sized piece |

# I License for StrokeRehab dataset

As part of research funded by the FUNDING AGENCY (NINDS, NLM), these data are made available from Dr. Heidi Schambra and her colleagues at NYU Langone Health. This repository of data has been made available thanks to funding from NIH grants R01LM013316 (CFG and HMS), K02NS104207 (HMS), and AHA postdoctoral fellowship 19AMTG35210398 (AP).

LICENSE AGREEMENT Downloading any of the provided data indicates your agreement to the following license agreement (License Agreement).

Use Agreement Permission is hereby granted, free of charge, to any recipient downloading a copy of these data and associated documentation files (the 'Data') to be used, subject to the terms of this License Agreement, for the recipient's non-commercial purposes, which shall be limited to

Table 12: Description of the activities performed by the stroke impaired patients in the cohort (2/2).

| Activity | Workspace | Target object(s) | Instructions |
|---|---|---|---|
| Drinking | Tabletop with water bottle and paper cup 18 cm to the left and right of midline, 25 cm from edge closest to patient | Water bottle (12 oz), paper cup (4 oz) | Open water bottle, pour water into cup, take a sip of water, place cup on table, and replace cap on bottle |
| Tooth brushing | Sink with toothpaste and toothbrush on either side of the countertop, 30 cm from edge closest to patient | Travel-sized toothpaste, toothbrush with built-up foam grip, faucet handle | Wet toothbrush, apply toothpaste to toothbrush, replace cap on toothpaste tube, brush teeth, rinse toothbrush and mouth, place toothbrush back on countertop |
| Moving object on a horizontal surface | Horizontal circular array (48.5 cm diameter) of 8 targets (5 cm diameter) | Toilet paper roll wrapped in self-adhesive wrap | Move the roll between the center and each outer target, resting between each motion and at the end |
| Moving object on/off a Shelf | Shelf with two levels (33 cm and 53 cm) with 3 targets on both levels (22.5 cm, 45 cm, and 67.5 cm away from the left-most edge) | Toilet paper roll wrapped in self-adhesive wrap | Move the roll between the center target and each target on the shelf, resting between each motion and at the end |

Table 13: List of anatomical angles. The system uses a rigid-body skeletal model to convert the IMU measurements into joint and segment angles. ‡ Shoulder total flexion is a combination of shoulder flexion/extension and shoulder ad-/abduction. *Thoracic angles are computed between the cervical vertebra and the thoracic vertebra. †Lumbar angles are computed between the thoracic vertebra and pelvis.

| Joint/segment | Anatomical angle |
|---|---|
| Shoulder | Shoulder flexion/extension<br>Shoulder internal/external rotation<br>Shoulder ad-/abduction<br>Shoulder total flexion‡ |
| Elbow | Elbow flexion/extension |
| Wrist | Wrist flexion/extension<br>Forearm pronation/supination<br>Wrist radial/ulnar deviation |
| Thorax | Thoracic* flexion/extension<br>Thoracic* axial rotation<br>Thoracic* lateral flexion/extension |
| Lumbar | Lumbar† flexion/extension<br>Lumbar† axial rotation<br>Lumbar† lateral flexion/extension |

non-commercial research and non-profit teaching or learning. Permission is furnished subject to the following conditions:

1. The recipient agrees to (i) recognize the contribution of Dr. Heidi Schambra, Dr. Carlos Fernandez-Granda, and NYU Langone Health as the source of the Data, and (ii) acknowledge The NYU Langone Health's receipt of funding pursuant to NIH R01LM013316 and K02NS104207, in all written, visual, or oral public disclosures concerning the recipient's use of the Data.

2. The Data was collected under a project funded by the NIH. The recipient agrees that the Data transferred under this Agreement may be subject to NIH Policies NOT-OD-17-109 and NOT-OD-20-075 (the 'Policy') and therefore is deemed under the Policy to be issued a Certificate of Confidentiality (https://grants.nih.gov/policy/humansubjects/coc/how-to-apply.htm). Accordingly, the recipient is required to adhere to the Policy and protect the

privacy of the individuals from whom the data were collected in accordance with the Policy and subsection 301(d) of the Public Health Service Act.

3. In the event the recipient's use of the Data gives rise to a publication, Drs. Schambra and Fernandez-Granda and colleagues may be included as co-authors to the extent her/their contribution to the publication meets generally accepted authorship criteria.
4. The Data may not be used for any commercial application, use, or purpose, without explicit written consent from NYU Langone Health.
5. The recipient will not use the Data, either alone or in concert with any other information, to make any effort to identify or contact individuals who are or may be the sources of Data. Should the recipient inadvertently receive identifiable information or otherwise identify a subject, the recipient shall promptly notify NYU Langone Health and follow NYU Langone Health's reasonable written instructions, which may include return or destruction of the identifiable information.
6. The recipient agrees to establish appropriate administrative, technical, and physical safeguards to prevent unauthorized use of or access to the Data. The recipient shall report to NYU Langone Health any unauthorized use or disclosure of the Data within 5 business days of when it becomes aware of such use or disclosure.
7. The recipient agrees to use the Data in compliance with all applicable laws, rules, and regulations, as well as all applicable professional standards and relevant institutional policies, including, if applicable, the completion of any IRB or ethics review or approval that may be required.
8. The Data may be used solely by the recipient and the recipient's scientists, faculty, employees, fellows, students, and agents whose obligations of use are consistent with the terms of this License Agreement.
9. Nothing in this License Agreement shall operate to transfer to the recipient any ownership or intellectual property rights relating to the Data.
10. If directed by NYU Langone Health or upon the conclusion of the recipient's research use of the Data, the recipient shall destroy the Data, and upon the request of NYU Langone Health, provide written certification of such destruction. If destroying the Data is infeasible, the recipient shall extend the protections of this License Agreement to such Data and limit further uses and disclosures of the Data to those purposes that make the destruction infeasible, for so long as the recipient maintains the Data.

**Liability Agreement** The Data is understood to be provided "AS IS". NYU Langone Health MAKES NO REPRESENTATIONS AND EXTENDS NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED. THERE ARE NO EXPRESS OR IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE, OR THAT THE USE OF THE DATA WILL NOT INFRINGE ANY PATENT, COPYRIGHT, TRADEMARK, OR OTHER PROPRIETARY RIGHTS. NYU Langone Health (Dr. Schambra's and Dr. Fernandez-Granda's research groups) are not liable for direct or indirect losses or damage, of any kind, which may arise through the use of the Data.

Except to the extent prohibited by law, the recipient assumes all liability for damages which may arise from its use, storage, disclosure, or disposal of the Data. Except to the extent prohibited by law, the recipient shall indemnify, defend and hold harmless NYU Langone Health from and against any claims, losses, damages and expenses caused, alleged to be caused, or arising from the recipient's use of the Data or breach of the terms of the License Agreement.

# J Datasheet for Dataset

## J.1 Motivation

**For what purpose was the dataset created?** The clinical motivation for the creating dataset has been explained in detail in Section 2. Essentially, the dataset was created to facilitate quantification of rehabilitation training in stroke patients. This capability would support the execution of a dose-response study to identify the optimal number of repetitions needed to maximally boost recovery after stroke.

**Who created the dataset (e.g., which team, research group) and on behalf of which entity (e.g., company, institution, organization)?** This dataset was created by Mobilis Lab, Department of Neurology, New York University. The principal investigator is Dr. Heidi Schambra.

**Who funded the creation of the dataset?** The research and dataset creation was funded by National Institute of Neurological Disorders and Stroke (K02 NS104207) and National Library of Medicine and the National Science Foundation (R01 LM013316).

## J.2 Composition

**What do the instances that comprise the dataset represent (e.g., documents, photos, people, countries)?** The instances capture the movement of the subjects performing activities of daily living like feeding, drinking from a glass, combing hair, etc. The movement is captured using video as well as IMU sensors attached to the upper body of the subjects. The detailed explanation of the acquired data is found in Section 2.

**How many instances are there in total (of each type, if appropriate)?** The dataset consists of 3,372 trials of rehabilitation activities performed by 60 stroke-impaired and healthy subjects. Cumulatively, they performed 120,891 functional primitives, which is more than existing benchmark datasets such as FineGym (32,697 annotated sub-actions), Breakfat (11,656 annotated actions), Jigsaws (1,701 annotated actions) and 50Salads (999 annotated actions).

**Does the dataset contain all possible instances or is it a sample (not necessarily random) of instances from a larger set?** Yes

**What data does each instance consist of?** Each instance consists of motion data of the subject. The motion data is captured either using videos or using IMU sensors. For videos, instead of providing the raw videos, we are providing the feature embedding for each frame of video. Refer to Section 2 and Section G.1 for details.

**Is there a label or target associated with each instance?** Yes, each time step of the trial is annotated with a functional primitive. The details about the labeling can be seen in Section 2 and Section 2.5. Approximately 2,700 human hours of manual effort was required to label the entire dataset.

**Is any information missing from individual instances?** No

**Are relationships between individual instances made explicit (e.g., users' movie ratings, social network links)?**

**Are there recommended data splits (e.g., training, development/validation, testing)?** Yes,

**Are there any errors, sources of noise, or redundancies in the dataset?** No, the labels are of high quality, with high inter-rater reliability (Cohen's kappa $\geq 0.96$). Details of the labeling procedure can be seen in Section 2.5.

**Is the dataset self-contained, or does it link to or otherwise rely on external resources (e.g., websites, tweets, other datasets)?** The dataset is self-contained.

**Does the dataset contain data that might be considered confidential (e.g., data that is protected by legal privilege or by doctor–patient confidentiality, data that includes the content of individuals' non-public communications)?** No.

**Does the dataset contain data that, if viewed directly, might be offensive, insulting, threatening, or might otherwise cause anxiety?** No.

**Does the dataset identify any subpopulations (e.g., by age, gender)?** The subpopulations are identified in the dataset based on the demography. The demographic distribution of the dataset can be seen in Table 1.

**Is it possible to identify individuals (i.e., one or more natural persons), either directly or indirectly (i.e., in combination with other data) from the dataset?** No.

**Does the dataset contain data that might be considered sensitive in any way (e.g., data that reveals race or ethnic origins, sexual orientations, religious beliefs, political opinions or union memberships, or locations; financial or health data; biometric or genetic data; forms of government identification, such as social security numbers; criminal history)?** No.

### J.3 Collection Process

**How was the data associated with each instance acquired?** Upper body motion was recorded while the subjects performed instances of functional primitives to execute activities of daily living. The activities included washing the face, applying deodorant, combing the hair, donning and doffing glasses, preparing and eating a slice of bread, pouring and drinking a cup of water, brushing teeth, and moving an object to horizontal and vertical target arrays. These activities are typically practiced in rehabilitation. See Appendix E.5 of the main paper for detailed descriptions of the activities. The patients performed five repetitions of each activity.

**What mechanisms or procedures were used to collect the data (e.g., hardware apparatuses or sensors, manual human curation, software programs, software APIs)?** For collection of kinematic data, upper body motion was recorded using nine Inertial Measurement Units (IMUs, Noraxon, USA) attached to the upper body, specifically the cervical vertebra C7, the thoracic vertebra T12, the pelvis, and both arms, forearms, and hands. These IMUs captured 76-dimensional kinematic features of 3D linear accelerations, 3D quaternions, and joint angles from the upper body (see Appendix E.2 and Appendix E.6 for details). As an additional feature, we included the paretic (stroke-affected) side of the patient (left or right) encoded in a one-hot vector, increasing the dimension of the feature vector to 77.

For collection of video data, upper body motion was synchronously recorded using two high definition cameras (1088 x 704, 60 frames per second; Ninox, Noraxon) placed orthogonally $< 2$ m from the patient. We also extracted frame-wise feature vectors from the raw videos. The detailed procedure to extract these feature vectors is mentioned in Appendix E.3.

**If the dataset is a sample from a larger set, what was the sampling strategy (e.g., deterministic, probabilistic with specific sampling probabilities)?** Not applicable.

**Who was involved in the data collection process (e.g., students, crowdworkers, contractors) and how were they compensated (e.g., how much were crowdworkers paid)?** Data was collected by a postdoctoral fellow and research coordinators who were employed full-time by Dr. Schambra.

**Over what timeframe was the data collected?** Data were collected over 2 years.

**Were any ethical review processes conducted (e.g., by an institutional review board)?** The study protocol was reviewed and approved by the institutional review board of New York University Grossman School of Medicine.

**Did you collect the data from the individuals in question directly, or obtain it via third parties or other sources (e.g., websites)?** Data were collected directly from the individuals.

**Were the individuals in question notified about the data collection?** The individuals went through a process of informed consent, in which they were informed about the process of data collection. The exact language in the consent form is as follows:

What will I be asked to do in the study? We will first test your movement ability. Based on these results, it is possible that you may not be asked to continue. If you qualify to be enrolled in the study, we will ask you to wear movement sensors and to be videotaped during activities that have you move your arms.

It will take about 15 minutes to set you up, and we will remove everything immediately after. We will place some of the motion sensors in elastic foam bands around on your arms, and in a headband around your head. Other motion sensors will be placed on your hands and back with double sided,

latex-free wig tape. Those on your back will go under your clothes. You can then move around freely afterward. Videotaping will also occur during the sessions.

The recordings will be used for analysis by the research team. The motion sensors recordings will not have any identifying information associated with them, but the videotaped recordings will record your facial features. This is unavoidable because you will make movements that will bring the arm hand near your face.

**Did the individuals in question consent to the collection and use of their data?** Once potential stroke subjects have been identified, the study team will notify the treating physicians (including the PI) that they have patients eligible to participate and request to directly contact the potential subject on their behalf. The study team will contact potential subjects by phone, email, or MyChart (attached: phone, SendSafe Secure email, and MyChart message scripts), and an additional verbal pre-screening will be undertaken (attached: waiver of documentation of consent). Phone and email conversations will be guided by IRB-approved scripts that explain the study procedures, risks, and benefits; and will serve to assess subjects' interest and administer pre-screening questions. Information that is collected will be immediately destroyed if a subject is determined to be ineligible or decides not to participate in the study. Subjects may also be contacted using IRB-approved language for follow-up communication (attached: follow-up/ thank you email template) after study completion. Furthermore, we will reach out to previously enrolled subjects and inquire about any members of their social circle that are interested in being contacted about our study (attached: recruitment email template). If a subject requests information regarding opting out of further recruitment for all research, he/she will be directed to contact research-contact-optout@nyumc.org or 1-855-777-7858.

**If consent was obtained, were the consenting individuals provided with a mechanism to revoke their consent in the future or for certain uses?** Consenting individuals were provided directions about how to revoke their consent in the future. The exact language in the consent forms is as follows: Can I change my mind and withdraw permission to use or share my information? Yes, you may withdraw or take back your permission to use and share your health information at any time. If you withdraw your permission, we will not be able to take back information that has already been used or shared with others. To withdraw your permission, send a written notice to the principal investigator for the study noted at the top of page 1 of this form. If you withdraw your permission, you will not be able to stay in this study.

**Has an analysis of the potential impact of the dataset and its use on data subjects (e.g., a data protection impact analysis) been conducted?** This analysis has not been conducted.

### J.4 Preprocessing/cleaning/labeling

**Was any preprocessing/cleaning/labeling of the data done (e.g., discretization or bucketing, tokenization, part-of-speech tagging, SIFT feature extraction, removal of instances, processing of missing values)?** For kinematic data, each IMU sensor captures 3D linear accelerations and angular velocities at 100 Hz. We used coordinate transformation matrices to convert angular velocities to sensor-centric unit quaternions, representing the rotation of each sensor on its own axes. In addition, proprietary software (Myomotion, Noraxon) generates 22 anatomical angle values using a rigid-body skeletal model scaled to patient height. See Section E.6 for a detailed description of these angles. The resulting 76-dimensional vector thus represents the kinematic features of 3D linear accelerations, 3D quaternions, and joint angles from the upper body. As an additional feature, we included the paretic (stroke-affected) side of the patient (left or right) encoded in a one-hot vector, increasing the dimension of the feature vector to 77. Each entry (except paretic side) was mean-centered and normalized separately for each task repetition in order to remove spurious offsets introduced during sensor calibration.

For video data, we extracted and released features from raw videos using the X3D model. The X3D model is a 3D convolutional network designed for efficiently performing the task of video classification. The model is pretrained on the Kinetic dataset, which consists of coarse actions like running, climbing, sitting, etc. Since the StrokeRehab dataset consists of elemental, sub-second actions, we fine-tuned the X3D model on the training set of StrokeRehab. For fine-tuning the model, we use video sequences as input and try to identify the primitive happening in the center frame of the videos. The fine-tuned model was then used to extract the frame-wise feature vectors from the raw videos.

**Was the "raw" data saved in addition to the preprocessed/cleaned/labeled data (e.g., to support unanticipated future uses)?** We release the "raw" sensor data. For video data, we do locally save the "raw" videos but release only the extracted video features.

**Is the software that was used to preprocess/clean/label the data available?** The code for preprocessing the data is available here: `https://github.com/aakashrkaku/seq2seq_hrar`

### J.5 Uses

**Has the dataset been used for any tasks already?** To facilitate the quantification of functional motions performed during stroke rehabilitation, we developed an approach that combines unobtrusive motion capture with automated identification. The StrokeRehab dataset which consists of labeled sensor and video data from stroke patients is used to train models to automatically identify and count functional primitives.

**Is there a repository that links to any or all papers or systems that use the dataset?**

1. Towards data-driven stroke rehabilitation via wearable sensors and deep learning [26]
2. Sequence-to-Sequence Modeling for Action Identification at High Temporal Resolution [27]
3. PrimSeq: a deep learning-based pipeline to quantitate rehabilitation training [47]
4. A Taxonomy of Functional Upper Extremity Motion [48]

**What (other) tasks could the dataset be used for?** StrokeRehab dataset can be have many potential use cases like using it for quantification of rehabilitation training by identifying and counting elemental motions. It could also be used for characterizing the normal motion done by the healthy subjects and using that as a guideline to assign data-driven impairment scores to stroke-impaired patients. The models, thus, build can keep a track of progress of stroke-impaired patients as they are go-through their rehabilitation training. Or, since the dataset annotates elemental-motions which are fundamental to any upper body movement, it could be used for pre-training the action-recognition models in a supervised or self-supervised manner which are then, used for a downstream task of motion identification. In this work, we focus on the first task of automatically identifying a sequence of actions from video or sensor data for quantifying rehabilitation.

**Is there anything about the composition of the dataset or the way it was collected and preprocessed/cleaned/labeled that might impact future uses?** No.

**Are there tasks for which the dataset should not be used?** No.

### J.6 Distribution

**Will the dataset be distributed to third parties outside of the entity (e.g., company, institution, organization) on behalf of which the dataset was created?** Yes

**How will the dataset will be distributed (e.g., tarball on website, API, GitHub)?** The data is available here: `https://simtk.org/projects/primseq` DOI: none

**When will the dataset be distributed?** The dataset has two modalities, IMU sensor data and extracted video features, and from two cohorts, stroke-impaired and healthy subjects. Currently, SimTK website has IMU data for stroke-impaired subjects. During the review period, remaining data would also be released. We have provided sample data from each cohort for reviewers' reference.

**Will the dataset be distributed under a copyright or other intellectual property (IP) license, and/or under applicable terms of use (ToU)?** Yes, the dataset will be distributed under the terms of a custom license agreement.

**Have any third parties imposed IP-based or other restrictions on the data associated with the instances?** No

**Do any export controls or other regulatory restrictions apply to the dataset or to individual instances?** No

### J.7 Maintenance

**Who will be supporting/hosting/maintaining the dataset?** The dataset is hosted at SimTK website - `https://simtk.org/projects/primseq`
SimTK is a free project-hosting platform for the biomedical computation community that:

- Enables to easily share the software, data, and models
- Tracks the impact of the resources that was shared
- Provides the infrastructure to support and grow a community around the project
- Connects the project to thousands of researchers working at the intersection of biology, medicine, and computations.

**How can the owner/curator/manager of the dataset be contacted (e.g., email address)?** The principal investigator of the study can be contacted at Heidi.Schambra@nyulangone.org.

**Is there an erratum?** No

**Will the dataset be updated (e.g., to correct labeling errors, add new instances, delete instances)?** No, this is not anticipated.

**If the dataset relates to people, are there applicable limits on the retention of the data associated with the instances (e.g., were the individuals in question told that their data would be retained for a fixed period of time and then deleted)?** No

**Will older versions of the dataset continue to be supported/hosted/maintained?** Not applicable

**If others want to extend/augment/build on/contribute to the dataset, is there a mechanism for them to do so?** They may create a supplemental dataset in SimTK that can be linked to StrokeRehab.