# Contributions overview

The following is a table detailing the main areas of responsibility for each group member. Though a single person was assigned to each task, there was total collaboration. **Everyone has familiarized themselves with all methods and packages, and tested and experimented with all products.** Take for example the LSTM, Simon was in charge of building; for getting the details hammered out, we discussed within the group, and all members ran the code themselves getting a feel for the pros and cons of the method.

| **Title** | **Person** | **Details** |
|---|---|---|
| Managing, delegating, organizing | Anton | To get everyone started Anton delegated tasks. He made a plan and overview of the project. |
| Naive integration | Anton | Gain overview, describe data flow, implement in python |
| Filtering | Adrian | Research and implement filters for IMU data processing |
| Custom loss functions for quaternions | Adrian | Quaternions have a symmetry property, i.e. q = -q. This must be taken into consideration when calculating the loss function. Three distance metrics in combination with two loss functions can be combined with the CustomLoss function implemented with pytorch. |
| Clustering as feature extension | Michael | Feature extension by clustering segments of data.<br>Mean, STD and FFT used as defining characteristics.<br>Silhouette score and elbow method used to determine number of clusters.<br>Kmeans, Agglo and Birch as clustering methods.<br>UMAP is used for inspection of clusters. |
| Data overview | Michael | Sensortype, sample rate, unit analysis, preprocessing and descriptions. |
| Linear network for quat estimation | Chris, Simon, Anton | Implement a linear neural network using pytorch/tensorflow. This was to get acquainted with the data, and brushing up on the deep learning libraries. |
| Linear network for position estimation | Chris | Build a Linear neural network |
| Transformer, end-to-end | Anton | End-to-end refers to the solution taking in IMU data and outputting position estimates. This was overly ambitious for my small cpu… |
| Linear network, guided by naive integration | Anton | Build a linear network, and supply analytical calculation of positions (obtained via naive double integration). |
| Data Loader | Anton, | Our data is split across many files, resulting from different |

|  |  |  |
|---|---|---|
|  | Simon | recordings. We needed a data-loading function, which took samples from many different recordings, and made sequences out of them. This turned into quite a massive function, with all kinds of optional parameters. |
| LSTM for quaternion estimate | Simon,<br>Anton,<br>Adrian | Implement an LSTM for predicting the change of the orientation (quaternion). We provide a sequence of IMU readings, along with an initial orientation. The model then predicts quaternion at the end of the sequence. |
| GRU and RNN for quaternion estimate | Adrian | Implement an RNN and GRU model for predicting the change of the orientation (quaternion). |
| Dynamic filter weighting with LSTM for quat estimation | Adrian,<br>Simon | Use filtered features in various combinations to improve on quad predictions |
| LSTM, GRU, RNN for position update | Simon | Implement the 3 architectures to predict the position of the walker given the IMU data along with the true orientations. |
| Hyperparameter optimization framework | Adrian | Build a framework for optimizing pytorch models using Optuna |

# Collaboration evaluation

**Key challenges:**

- Being a group of 5, some with full or part-time work, we had some difficulty organizing physical meetings.
- Division of labor was not clear until we had gained a solid overview.
- We spent too long on quaternion estimation, thus not leaving time to tie together the different networks for an end-to-end solution.

**Time management**: We initially knew it would be a time consuming project,

**Division of responsibility**:
At each meeting, we discussed what tasks were assigned to each person to make sure everyone had something to do, and that our individual work was helping us progress on the ensemble target.

**Working environment**: Coding together on Discord, at exercise classes and individually at home with sharing, discussion and realignment meetings. The methods satisfied all group members individual preference for learning and adhered to everyones specific worklife needs.

**The red-thread**:
Early on in the project we established a timeline and attack-vector. This was great for guiding us along the way.

**Code sharing**:
Code sharing was done through a shared GitHub. We got this set up quickly, and we were able to work together in an effective manner.

**Utilities**:
We build a rich library of functions for requesting samples from our dataset. Additionally, we built a library for analytic processing of IMU data. This proved very helpful, as we did not have to implement these things again and again.