NORTHWESTERN UNIVERSITY

Responses of Somatosensory Area 2 Neurons to Unconstrained Three-Dimensional Upper Limb Movements

A DISSERTATION

SUBMITTED TO THE DEPARTENT IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

for the degree

MASTER OF SCIENCE

Field of NEUROBIOLOGY

By

Qiwei Dong

EVANSTON, ILLINOIS

June 2021

Abstract

Previous studies have shown that neurons in the arm region of area 2, a subregion of the primary somatosensory cortex (S1), has a representation of proprioception during upper limb movements. An effective way to investigate what types of external covariates these neurons represent is to build different neural encoding models and evaluate their performances. A previous study has shown that Generalized Linear Model (GLM) can better predict neural responses when using a neural encoding model incorporating elbow kinematics with hand kinematics, implying that area 2 neurons might encode whole-arm kinematics. Although, the results also showed that the model with whole-arm kinematics only performed a little bit better compared to the model with only hand kinematics. I assume that the current experiment paradigm, the Random Target 2D (RT2D) task, would increase the correlation between hand and elbow movements with limited workspace and simple movements, making it harder to differentiate the performance between the two models. It is thus necessary to improve the experiment paradigm and the data analysis tools, so that we can increase the monkey arm's workspace and break the high correlations between elbow and hand movements. If larger performance differences appear when comparing the neural encoding performances between the two models using new experiment paradigm, it would further support the idea that area 2 neurons encode whole-arm kinematics, instead of just hand-endpoint kinematics.

Here, I propose a novel experimental paradigm to obtain monkey's upper limb kinematics during unconstrained movements in a three-dimensional workspace and see if this new setup makes a difference in terms of the experiment results compared to the models using previous experiment paradigms. I found that Random Target 3D (RT3D) task did not completely decorrelate the kinematic relationships between elbow and hand. Also, the prediction performances between the hand-only model and the whole-arm model were not greater for the 3D version of the reaching task.

Table of Contents

Lists of Tables, Illustrations, Figures, or Graphs

Introduction

Proprioception is a covert, yet fundamental type of bodily neuromuscular sense that exists in our daily lives. Proprioception falls under the category of Somatosensation, and covers the sense of limb position, movement, and associated forces (Ager et al., 2020). The sensory information of proprioception is generated by three types of proprioceptors, joint receptors, muscle spindles and Golgi tendon organs. This information will be eventually fed to area 2 of primary somatosensory cortex (S1), the proprioceptive representative area in the cerebral cortex. Previous studies have proved proprioception's importance in making coordinated movements (Ghez & Sainburg, 1995; Gordon, Ghilardi, & Ghez, 1995; Sainburg, Poizner, & Ghez, 1993; Sanes, Mauritz, Evarts, Dalakas, & Chu, 1984). Despite its importance, the representation of proprioception in the brain during natural movements has not been explicitly addressed, compared to other senses, such as vision and motion (Chowdhury, Glaser, & Miller, 2019).

A previous experiment studied the S1 neural activity during active arm reaching movements suggested that the proximal arm representation within area 2 is a brain area that is likely in charge of reach-related proprioception (Prud'homme & Kalaska, 1994). In the experiment, a hand-only model was used to model the limb movement-related neural activities in S1. This model assumes that the neural activities relate only to the Cartesian coordinates of hand position and velocity. They used a center-out reaching task, which asked the monkey to make whole arm reaching movements to visual targets in eight fixed

directions (Prud'homme & Kalaska, 1994). This task highly correlates the elbow and hand kinematics together, which would result in similar predictions of neural activity from different models. A subsequent study has shown that area 2 represents the state of the whole arm during reaching, by comparing the neural encoding accuracy of a hand-only model and a new model with additional elbow kinematics (whole-arm model) on a two-workspace task (Chowdhury et al., 2019). A two-workspace task, improved from the center-out reaching task, asks the monkey to reach to randomly generated targets presented in two different workspaces of the same height. This allows the monkey to make less stereotypical movements, aiming to decorrelate the upper limb kinematics, and being able to more reliably separate between models (Chowdhury et al., 2019).

The results from these reaching studies have provided a solid background for the basic relationships between reaching movements and neural responses from area 2. However, some limitations still remain in the previous experiment. Even though the purpose of the two-workspace task was to decorrelate the monkey's elbow and hand kinematics, the fact that the monkey's hand is still reaching in the 2D space still retains a lot of correlation between the two joints. When comparing the model prediction performances between a hand-only model and a whole-arm model with a dataset having high correlations between hand and elbow kinematics, both models will generate similar performances, since adding elbow kinematics highly correlated to hand kinematics will not provide extra useful information to the whole-arm model. Additionally, the motion capture system used in the previous study requires colored markers to be painted on the monkey's skin prior to each

session. This adds extra manpower and time to conduct the experiment and would likely influence the experiment result by stressing the monkeys.

Here, we designed a new experiment paradigm that overcomes these limitations, evaluated the performance of the experiment paradigm and the data recording pipeline, and tested both GLM models' neural encoding accuracy on the random target 2D task (RT2D task, the original experiment paradigm of the two-workspace task) and the new experiment paradigm, the random target 3D task (RT3D task). The RT3D task increases the workspace of the arm from a plane to roughly a semi-ellipsoid. It is designed to decorrelate the movements of the elbow and hand.

I expect to see that the whole-arm model performs better than the hand-only model in both RT2D task and RT3D task, if the argument that area 2 encodes the whole arm kinematics is correct. However, the whole-arm model is expected to perform better in the RT3D task, relative to the RT2D task, due to the decreased correlation between the joints, which would provide additional useful kinematical information to the whole-arm model. Alternatively, the whole-arm model might not perform better than the hand-only model in both tasks, possibly caused by the reason that area 2 does not really encode whole-arm kinematics, but only the hand kinematics instead.

Methods

A synchronized recording pipeline is used in this study, shown in figure 1. Neural signals were collected by the electrode array implanted in the monkey's primary somatosensory cortex, amplified by the amplifier, and recorded by the computer after being further processed by the neural signal acquisition system. The cameras were synchronized by the neural signal acquisition system with electrical pulses at 25Hz.

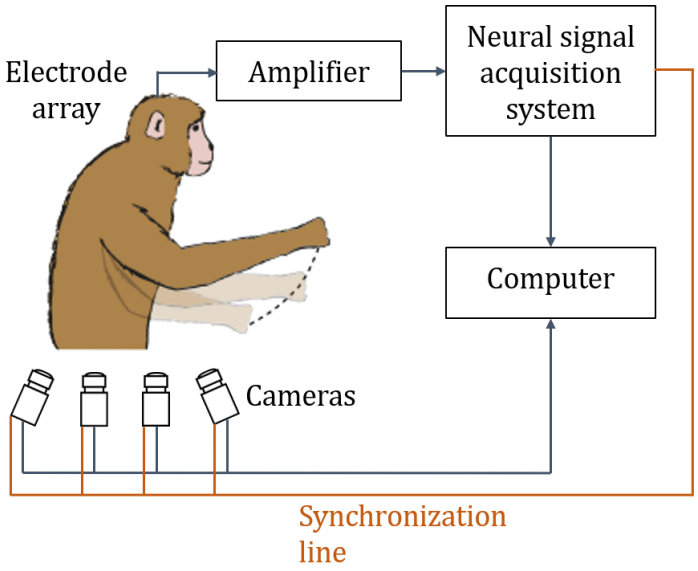


Figure 1: Recording pipeline for synchronized neural signal and video recordings. Neural signal acquisition system takes in amplified neural signals collected from the electrode array and sends pulses through the synchronization line to the cameras. For each pulse, the cameras acquire one frame, and send to the computer.

The recorded neural data was then sent for further signal processing and spike sorting, shown in Figure 2, to acquire single unit spike timing data. The video data, on the other hand, was processed by DeepLabCut (DLC) to track the landmarks on the limb of the monkey, and then reconstructed in the 3D space. Now the video data has been transformed to 3D kinematic data. The processed neural and kinematic data were then sent to the encoding model, synchronized with each other, for kinematical analysis.

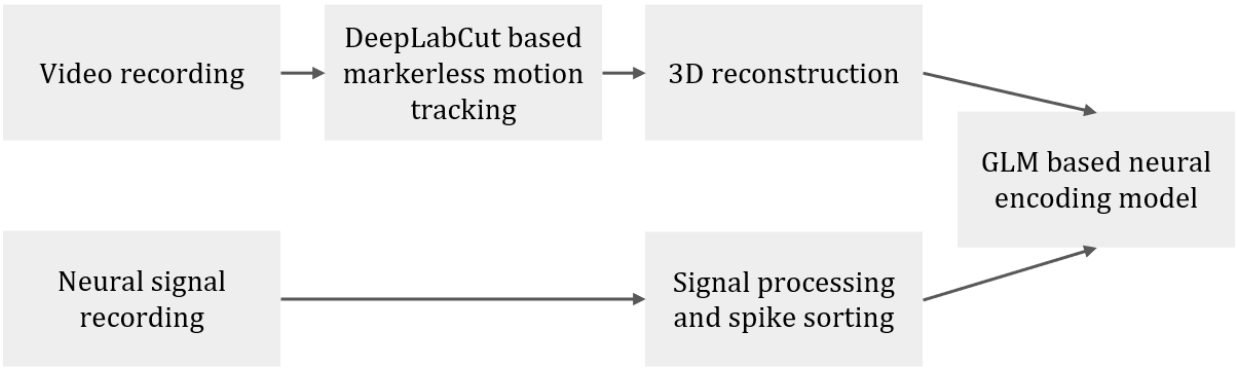


Figure 2: Neural and video data processing pipeline. Video data goes through DLC motion tracking and 3D reconstruction. Neural data goes through spike sorting. Both datasets will be sent to GLM model.

Behavioral tasks

Two rhesus macaques, weigh 10-12 kilograms were trained to perform two behavioral tasks, a random target 2D task (RT2D), and a random target 3D task (RT3D), as shown in Figure 4. In the RT2D task (Figure 3-A), the monkey was positioned in a primate chair,

holding a handle on a two-link planar manipulandum that moves horizontal to the floor. The monkey used this manipulandum to control a cursor on a LED screen in front of the monkey to reach to a target randomly appearing on the screen (London & Miller, 2012). A successful reach for this experiment is defined as: the monkey controls the cursor from a random place on the screen to reach to the target shown on the screen. A successful trial for this experiment is defined as: the monkey made successful reaches to 4 targets in a row and held on the target for at least 0.15 seconds. The monkey received a water reward after each successful trial, and if it failed for one target in this period, the trial recounts from the beginning. The targets are 5.5cm wide squares, and the side length of the workspace for this experimental paradigm was 20 cm on the X axis, and 20 cm on the Y axis.

In the random target 3D task (Figure 3-B), the monkey was seated in the same chair. But instead of grasping a handle, the monkey was expected to reach to the tip of a stick provided by the experimenter instead of the target on the screen. The tip of the stick is 1 cm wide. The monkey will receive a water reward every 2-4 times it reached to the target consecutively. In this experiment paradigm, I was not separating experiments in terms of trials, but only in terms of whether the monkey is actively reaching in a period of time or not. In terms of the size of the workspace, the side length for this RT3D task is around 20 cm on the X axis, 30 cm on the Y axis, and 30 cm on the Z axis. RT3D task allows the monkey to generate a dataset with a more natural representation of its neural responses to reaching activities, providing a path to dig into more details about S1 neural responses to

monkey's upper limb movements. For both tasks, the monkey was only allowed to use their right arm to reach for targets. Each task lasted around 15 minutes long.

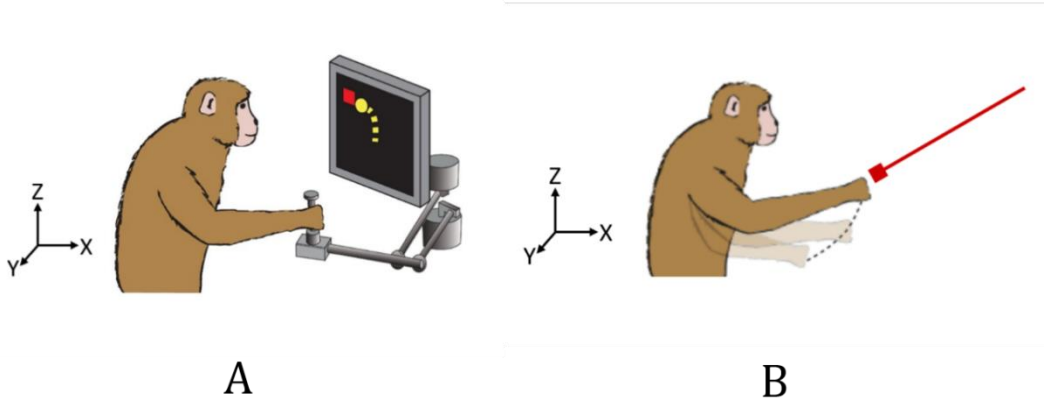


Figure 3: Example figures for RT2D task and RT3D task. (A) Figure of RT2D task. Monkey controls a cursor on screen (yellow) with a two link manipulandum to reach to visually presented targets (red). (B) Figure of RT3D task. Monkey reaches to a target provided by the experimenter.

Neural recordings

I used Utah arrays (Blackrock Microsystems) to record neural signals from the arm representation within somatosensory cortical area 2 from two Rhesus macaques (Monkeys C and H) (Figure 4-A). The arm representation in area 2 was determined intra-operatively through examination of the gross anatomy, stimulating primary motor cortex to find proximal arm representation, and recording from the surface of the post-central gyrus to identify proximal arm responses. In each surgery, the specific implant site was chosen to

maximize the arm representation while avoiding cerebral vasculature (Weber et al., 2011). The arrays used were 4-mm square, with 96 1-mm-long iridium-oxide coated electrodes (Figure 4-B).

The neural spike data was recorded at 30 kHz, and then sent to Plexon Offline Sorter for spike sorting. Spikes were detected using a threshold set at -4.5 times the root-mean-square (RMS) amplitude of the signal on each channel, and the time stamp and a 1.6ms snippet of each spike surrounding the threshold crossing were detected. These sorted spikes are then grouped into clusters by several features. In my case, I mainly used the 1st and 2nd principal components, and non-linear energy as references for sorting a neuron unit out. After clustering, the spikes are sorted into units based on their shape, amplitude, spiking frequencies across time, and interspike intervals. The characteristics of the neural data for monkey C and H are different. Monkey C has less units and less spikes, but also has less noise recorded. Monkey H on the other hand, has more units and spikes, but also has more noises. The sorted spike data will then be binned to 50ms firing rate data and synchronized with the kinematic data.

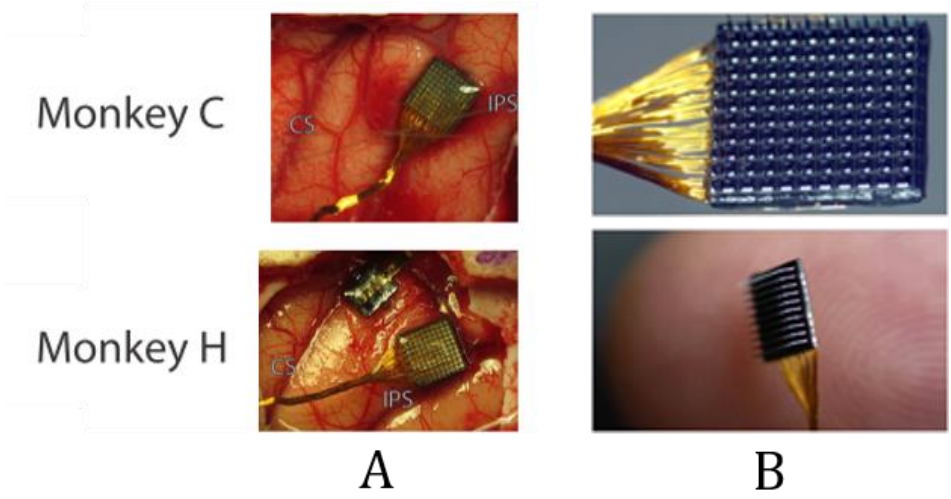


Figure 4: Array locations and example Utah array picture. (A) Locations of Utah arrays implanted in area 2 of Monkeys C and H. The Utah array is implanted in between the Central Sulcus (CS) and Intraparietal Sulcus (IPS), very close to the IPS. (B) Example pictures of Utah array.

Markerless motion tracking

Hardware setup

I used 4 synchronized cameras to record the monkeys' right arm movements for both RT2D and RT3D tasks. For the settings of these cameras, I recorded with a resolution of 1280*1024, a color format of RGB24, and a frame rate of 25fps. To synchronize the videos, a synchronization input is used to record a frame. The camera records a frame if the synchronization channel receives a pulse. The pulse is sent by the Cerebus™ Neural Signal

Processor from the Blackrock Microsystems, which also acquires the neural signal recordings.

All 4 cameras were positioned on the right side of the monkey, mounted on a metal frame. The two lower cameras were intended to optimally record the monkey's shoulder and elbow movements, and the two upper cameras to record the monkey's wrist and hand (Figure 5). To obtain these movements, we chose to record and track several anatomical landmarks on the monkey's right arm. The landmarks were chosen to be the same as the previous experiment's (Chowdhury et al., 2019). While positioning the cameras, I made sure that a landmark can be seen by at least 2 cameras. This does not mean that these cameras cannot see the other landmarks at all during recording, it is just that these cameras are expected to record their designated landmarks more consistently and accurately, compared to the landmarks they were not optimized for.

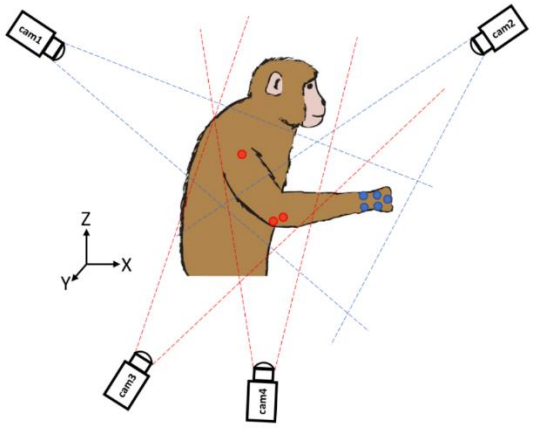


Figure 5: Monkey's right arm workspace in the camera's field of view. The two lower cameras (3 and 4) are positioned to track the proximal (red) landmarks, and the two upper cameras are positioned to track the distal (blue) landmarks.

Training deep neural network for extracting virtual landmarks

I am using an application called DeepLabCut (DLC) (Mathis et al., 2018), to extract estimated monkey arm poses from the experimental videos. DLC is a markerless pose estimation software that extracts animal poses. Given a set of hand-labeled training images, DLC trains a robust feature detector that detects landmarks in the experimental videos.

To acquire ground truth data to train the model, DLC extracts a set of frames from the experiment videos, asks the experimenter to label these frames with the positions of the landmarks, and divides the labeled frames into a training set and a testing set (Figure 6-A, B). In my experiment, I use DLC's k-means method to extract these frames, based on the k-means clustering algorithm. K-means clustering algorithm is a vector quantization algorithm that aims to separate the incoming data into k clusters, with each data point belonging to a cluster closest to it. DLC's K-means method treats the video as an array of vectors, with each frame being one vector, and clusters the array with k-means algorithm. Several frames will then be extracted from each cluster separately, to make the ground truth frames as different from each other as possible.

DLC uses the labeled frames to supplement a more generally trained neural network, ResNet(He, Zhang, Ren, & Sun, 2016), that was pretrained on ImageNet (Deng et al., 2010)) (Figure 6-C), with several hundred thousand iterations. ResNet is a convolutional neural network, and ImageNet is a large image database used in visual recognition and object detection. The definition of finishing one iteration in this context is that the neural network

has processed the training dataset, calculated the cost function, backpropagated to the whole network and adjusted all the weights.

After training, DLC outputs a dataset estimating where the landmarks are in the entire video (Figure 6). To accomplish this, DLC analyzes all the pixels on a frame, estimates a likelihood value between 0 and 1 that it represents a given landmark (Figure 6-D). Then, it outputs the location of the pixel with the highest likelihood, together with the highest likelihood value, of that landmark on that frame (Figure 6-E). If the landmark is not occluded on a frame and the model is well trained, DLC model will output the landmark position and a likelihood value close to 1. If the landmark is occluded on a frame, DLC will still output a landmark position for that frame, but with a likelihood close to 0 (Figure 8).

To evaluate the model performance, DLC calculates the error in the test set in terms of the mean Euclidean distance between the hand-labeled landmark positions and the landmark positions inferred by DLC. Only the landmarks with a likelihood higher than a selected threshold will be considered when calculating model performances.

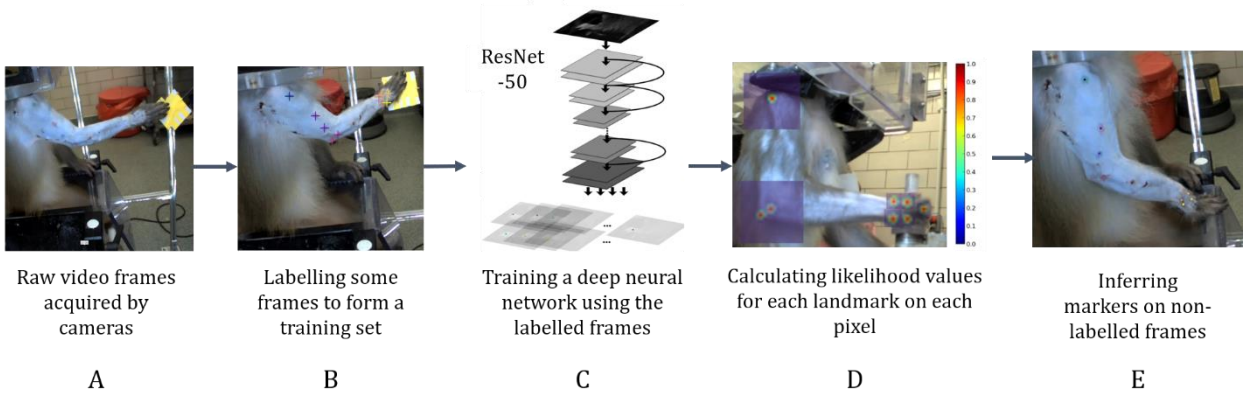


Figure 6: Pipeline for using the DLC toolbox to infer landmarks. (Mathis et al., 2018). (A) Input raw videos into DLC pipeline. (B) Label landmarks on ground truth frames selected by DLC for a training set. (C) Train a deep neural network to predict the landmark positions based on the training frames. (D) For all the unlabeled frames, use the deep neural network to calculate how likely each landmark is on a pixel. (E) Predict the landmarks' position on all the unlabeled frames based on the likelihood value.

I chose to track 8 anatomical locations as DLC landmarks. The anatomical locations I used are: the greater tubercle of the humerus ("shoulder1"), the head of radius ("elbow1"), the olecranon articulating with olecranon fossa of humerus ("elbow2"), the styloid process of radius("wrist1"), the styloid process of ulna ("wrist2"), the proximal tip of the 2nd metacarpal bone ("hand1"), the distal tip of the 3rd metacarpal bone ("hand2"), and the proximal tip of the 4th metacarpal bone ("hand3") (Figure 7).

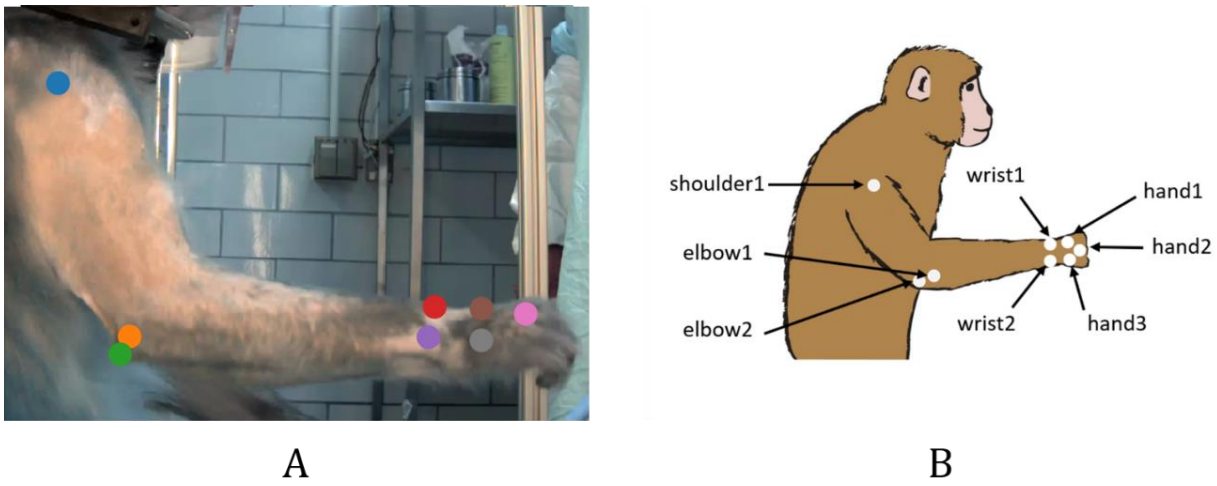


Figure 7: Landmark positions chosen to track to reconstruct arm kinematics. (A) Example frame of the landmark positions labeled on monkey's right arm. (B) The names of these landmarks.

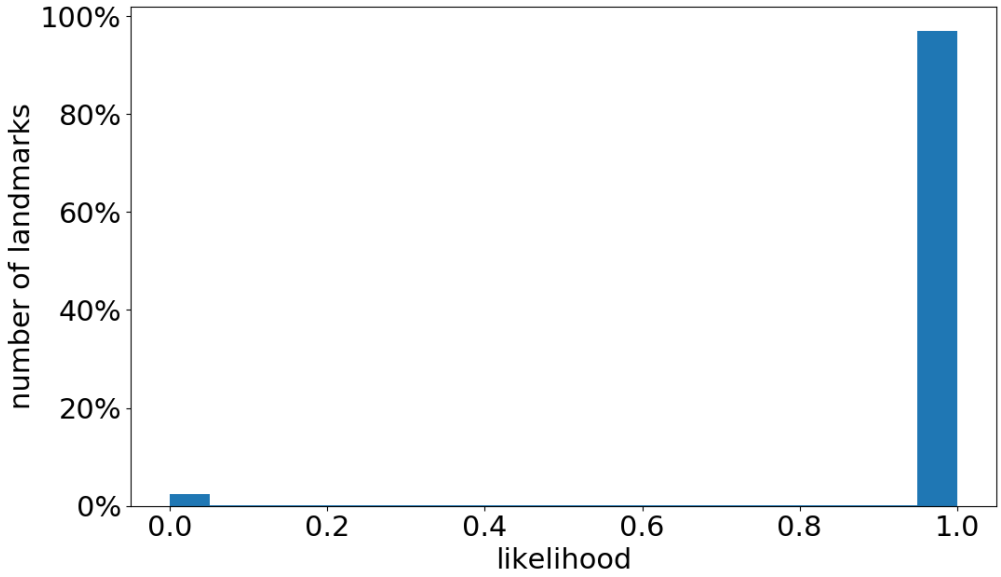


Figure 8: Histogram of percentage of landmarks with likelihood from 0 to 1. 97% of the landmarks had a likelihood value higher than 0.9. There were virtually no likelihood values between these limits.

Calibration and 3D reconstruction

I am using a program developed by Min-Young Park to accomplish the 2D to 3D reconstruction based on Direct Linear Transformation (DLT) methods (Bardsley, n.d.). The process consists of two major steps: calibration (both intrinsic and extrinsic) and 3D reconstruction. Calibration requires the experimenter to record short videos from each camera of a ChArUco calibration pattern (Figure 9) held in a variety of positions and orientations. Intrinsic calibrations calculate the cameras' individual characteristics, such as focal length, optical center, skew and distortion. Extrinsic calibrations describe the cameras' position and orientation relative to each other and the world. The ChArUco calibration pattern (Figure 9), one of the standard patterns for camera calibration used by Open-Source Computer Vision (OpenCV), is the ground truth reference for these intrinsic and extrinsic calibrations.

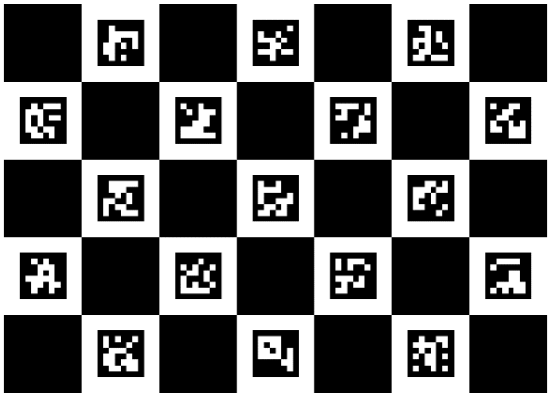
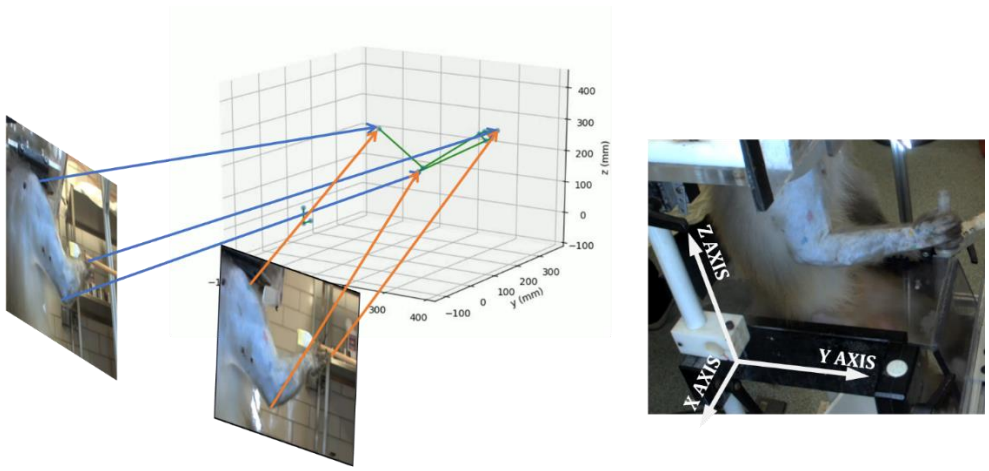


Figure 9: An example of the ChArUco board. A ChArUco board is a calibration pattern combined with a chessboard pattern and an ArUco pattern, allowing calibration with only partial view of the ChArUco board.

The next step, 3D reconstruction, is to project the landmarks in the camera spaces to the 3D space. The main concept of 3D reconstruction, as shown in Figure 10, is that if a landmark can be recorded in 2 or more camera spaces, we could project (shown by the blue and orange lines) this landmark from the camera spaces back to the 3D space based on the intrinsic and extrinsic calibration parameters, and get the 3D coordinates of this landmark.

After 3D reconstruction, we established a set of standard axes as our "room coordinates" (Figure 10-B), which the 3D reconstructed landmark data would be then rotated to. The positive X-axis points to the right of the monkey, the Y-axis points to the front of the monkey, and the Z axis points upward.



A B

Figure 10: Example figures for 3D reconstruction and rotation. (A) Example figure of reconstructing landmarks from 2 camera spaces to 3D space. (B) Reference figure of the axes that the landmarks in the 3D space will be rotated to.

Reprojection error can be determined by calculating the distance between camera recorded (DLC inferred) landmark position, and 2D reprojected landmark position (Figure 11-B). I calculated the reprojection error for each landmark in each camera view in the dataset. An example of reprojection error can be seen in Figure 11-A. The offset between the "X" points and "+" points of the same color is the 2D reprojection error of each landmark.

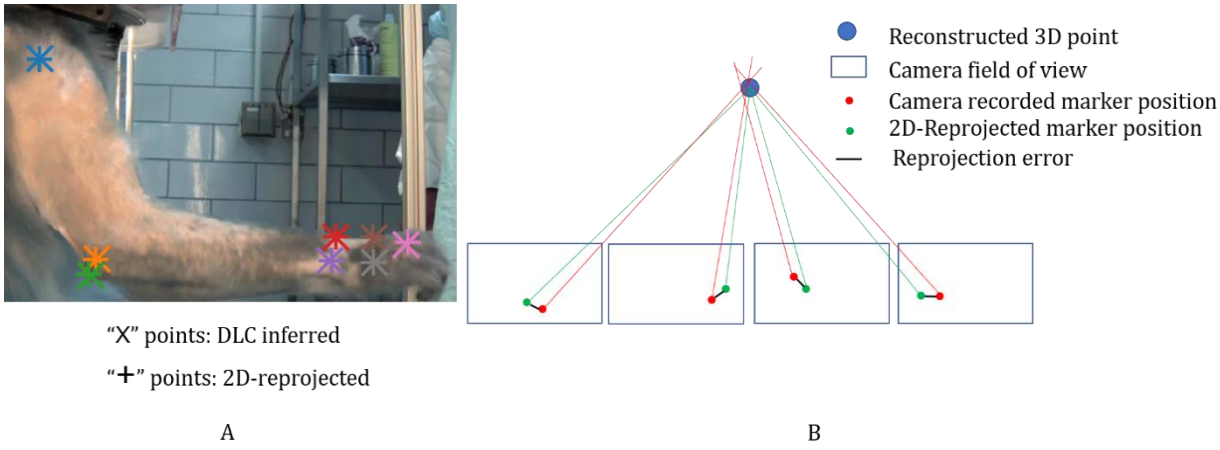


Figure 11: Example figures of 2D-reprojection error. (A) Example figure of the distance between a DLC inferred landmark position and a 2D-reprojected landmark position. (B) Example figure of the cause of 2D reprojection error.

Generalized linear model application

To determine what type of information about limb movements is encoded by the neural activity, I used generalized linear models (GLM) to test how well neural activity could be predicted by different sets of kinematics covariates. A GLM is a flexible version of linear

regression, capable of fitting data with non-Gaussian error distributions. In constructing neural encoding models, GLM works well, since it is assumed that neural firing rates follow a pattern of Poisson distribution, instead of a Gaussian distribution. Since GLM still holds the structure of a linear model, the interpretability of a linear regression model is retained.

Landmark position and velocity are being used as kinematic inputs for this model (Figure 12). The dataset was segmented so that only the portions when the monkey was actively reaching to targets were retained. For the hand-only model, I used hand position, velocity as inputs, and for a whole-arm model, I used hand position and velocity, elbow position and velocity. From the three landmarks on the hand, and two on the elbow, I chose one hand landmark and one elbow landmark.

The goodness-of-fit performances of these two models are evaluated using pesudo-R^2 (pR^2). The formulation of pR^2 I am using is based on the comparison between the deviance of the full model and a 'null' model, that is, a model that only predicts the overall mean firing rate(Chowdhury et al., 2019). The pR^2 metric ranges from - ∞ to 1, with 1 meaning a perfectly fit model, and 0 meaning a model that fits only as well as the "null" model.

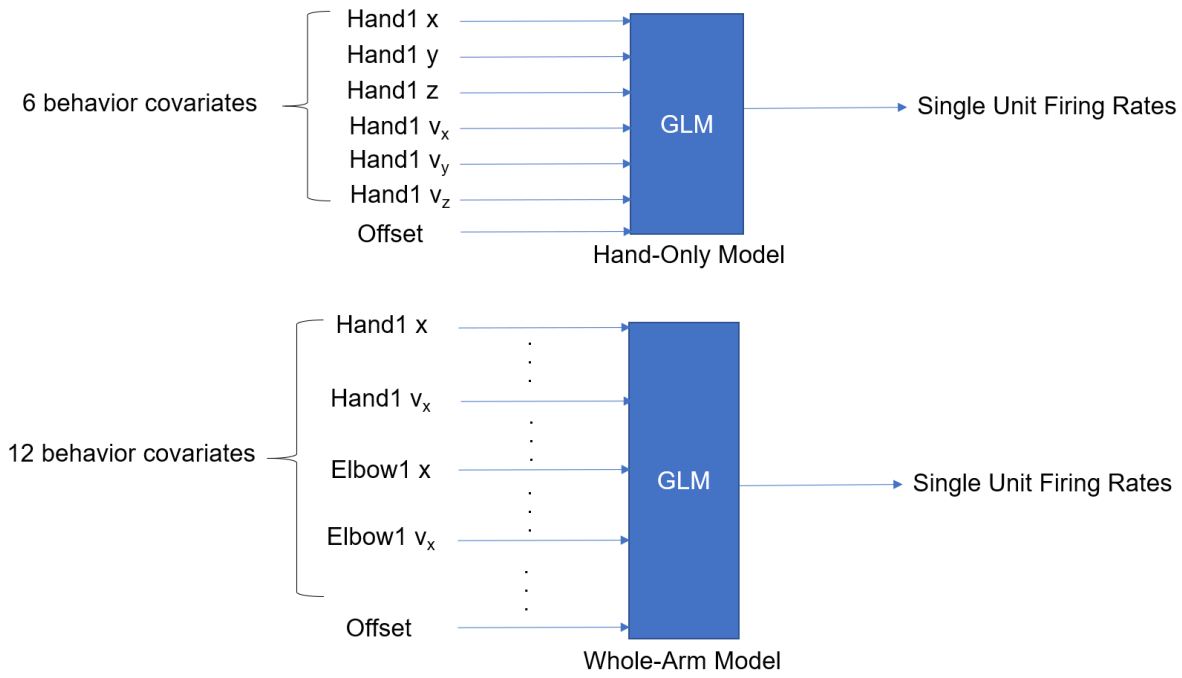


Figure 12: GLM input and output data for a hand-only model and a whole-arm model. The behavior covariates (X, Y, Z position and velocity) are the model input. The single unit firing rates is the model output.

Results

I recorded neural signals from two monkeys (C and H) using Utah multi-electrode arrays (Blackrock Microsystems) implanted in the arm representation of Brodmann's area 2 of S1. I also recorded kinematic data for both monkeys' right arm. Both monkeys were trained to perform the RT2D task and RT3D task while seated in a monkey chair. For each single session, I recorded 15 minutes for each task. Five datasets were collected for each task for monkey H, and four for monkey C. For this thesis, I am only going describe the analysis of one representative dataset for each monkey.

A significant difference between the two monkeys that I am using for this study is the tattoos on monkey H (Figure 13-B). In an earlier experiment, our lab used a tracking system based on Microsoft Kinect, and as part of the system, colored tattoos like these were used for the tracking system to locate the landmarks (Chowdhury et al., 2019). Since this monkey is necessary for this experiment, I decided to include the data recorded by this monkey in the experiment, while at the same time looking into the question that whether tattoos would affect the performance of this new DLC tracking system that we are using. Monkey C does not have tattoos.

A (Monkey C) B (Monkey H)

Figure 13: Reference picture of tattoos on monkey's arm. (A) Example picture of a non-tattooed monkey's arm. (B) Example picture of a tattooed monkey's arm.

DeepLabCut tracking performance

I tested the DLC's model prediction error on monkey C, (without tattoos) with different numbers of training iterations (Figure 14-A). To train this model, I used 829 hand-labeled frames, with 95% (788) of them for training, and 5% (41) for testing. The training dataset was taken from 6 experiment sessions from 3 days. DLC converged on both training datasets and testing datasets at roughly 500,000 iterations. The training set's RMSE plateaued at around 2.1 pixels, and the testing set's RMSE stayed at around 4.9 pixels.

I then tested the DLC's model performance on the same monkey, with increasing numbers of training frames (Figure 14-C), while using the same test set. I tested the performance of

7 models with 41 (5%), 82 (10%), 165 (20%), 248 (30%), 331 (40%), 414 (50%), and 746 (90%) frames of the training dataset, and 10% of the whole dataset as the test dataset. 600,000 iterations were trained on each model. Training error plateaued at 2.2 pixels with only 82 training frames. Testing error on the other hand, doesn't seem to plateau even with 746 training frames, as it's still slowly going down from RMSE of 5.1 pixels to 4.9 pixels.

Similar tests were done on the tattooed monkey, monkey H. I used 560 labeled frames, with 532 of them being the training set, and 28 of them as the test set. The tattooed model converged significantly faster than the non-tattooed model, and had a overall lower RMSE error at around 4.0 pixels for test errors, and 2.1 pixels for training errors (Figure 14-B).

I also tested increasing number of training frames with 600,000 iterations (Figure 14-D). Training error plateaued at 2.1 pixels with only 28 frames, while the testing error went down progressively to around 3.3 pixels with 282 training frames.

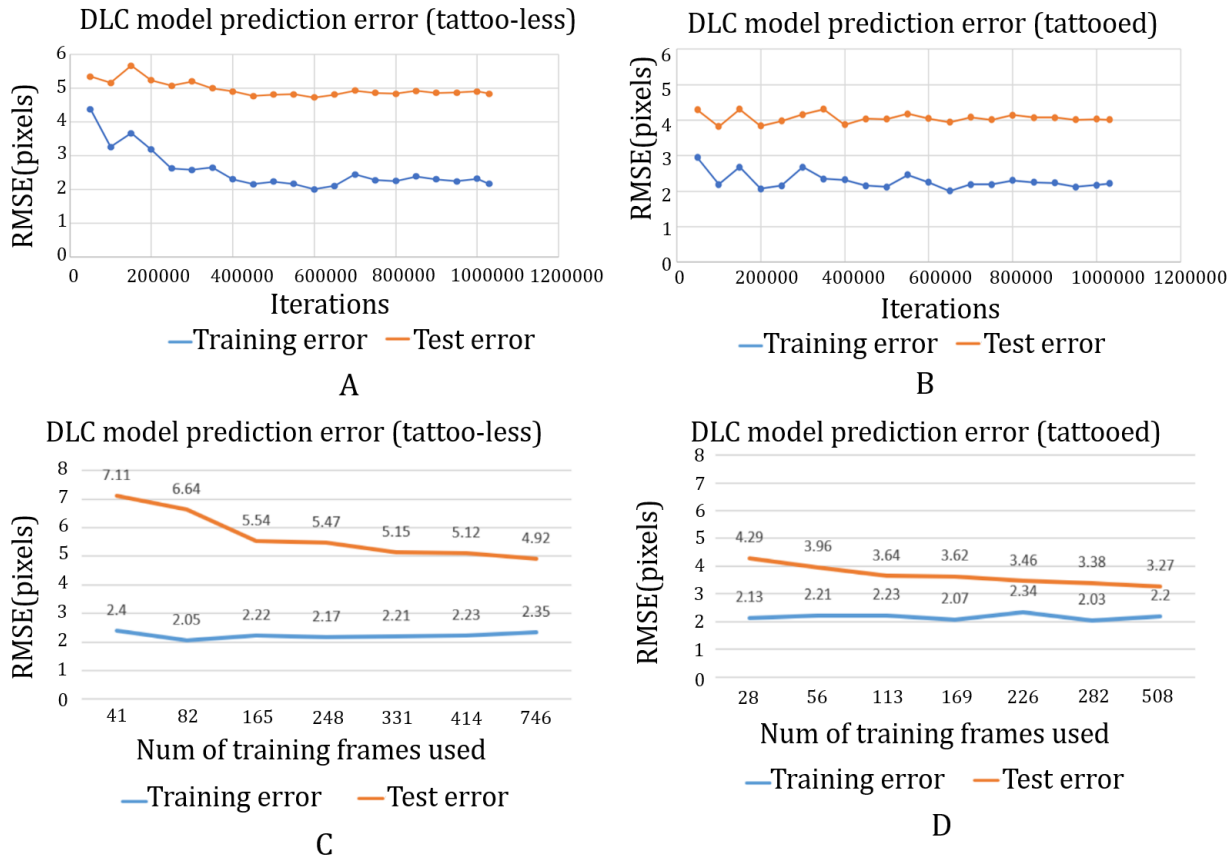


Figure 14: DLC model prediction error on non-tattooed and tattooed monkeys. (A-B) Prediction error related to number of training iterations on monkey C (A) and monkey H (B). (C-D) Prediction error related to number of training frames on monkey C (A) and monkey H (B).

To determine whether the different prediction performance between tattooed model and non-tattooed model was caused by particular landmarks, I calculated errors for each type of landmark (shoulder, elbow, wrist and hand) in both monkey C (non-tattooed) and monkey H (tattooed) (Figure 15). The mean testing error for monkey C's shoulder, elbow and wrist landmarks were around 4 pixels, while the hand was 6 pixels. The mean testing

error for all the monkey H's landmarks were around 4 pixels. These results indicated that the overall training error for shoulder, elbow and wrist landmarks are roughly the same, with the non-tattooed monkey having a wider distribution of errors. The mean training error for hand landmarks for a non-tattooed monkey is higher than that of a tattooed monkey, also with a wider error distribution.

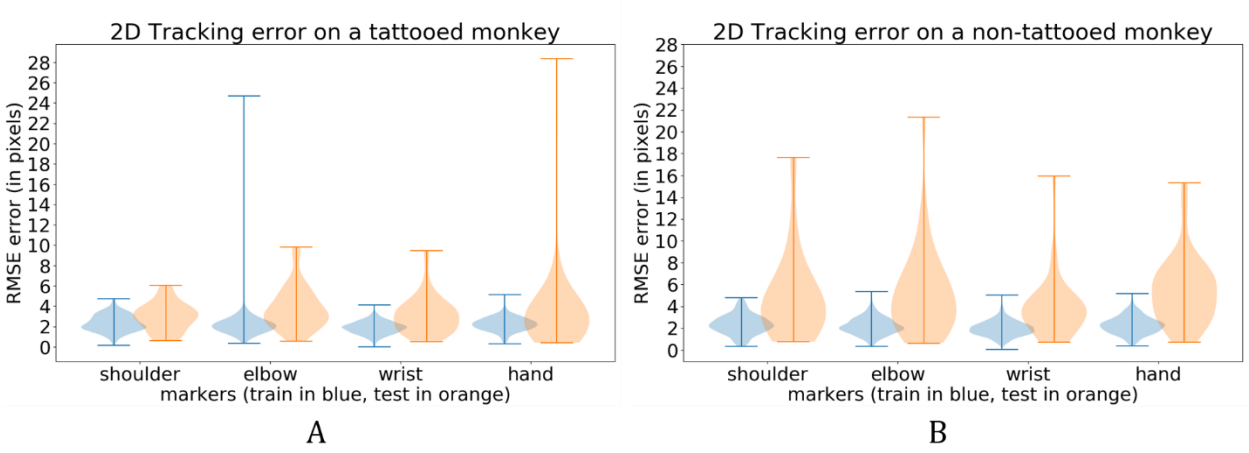


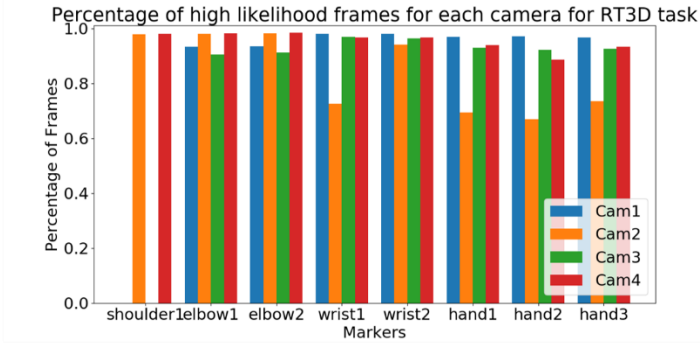
Figure 15: DLC training (blue) and testing (orange) prediction performance for each type of landmark (shoulder, elbow, wrist and hand) on a non-tattooed (figure A) and a tattooed monkey (figure B). Y axis represents RMSE in number of pixels.

I concluded that the test error of a DLC model aiming to track landmarks on a non-tattooed monkey's upper limb can get to 5.1 pixels by training the model with 400 training frames and 600k iterations. On a tattooed monkey, the DLC model can improve its performance, reducing the test error to 3.3 pixels by training the model with only 280 training frames.

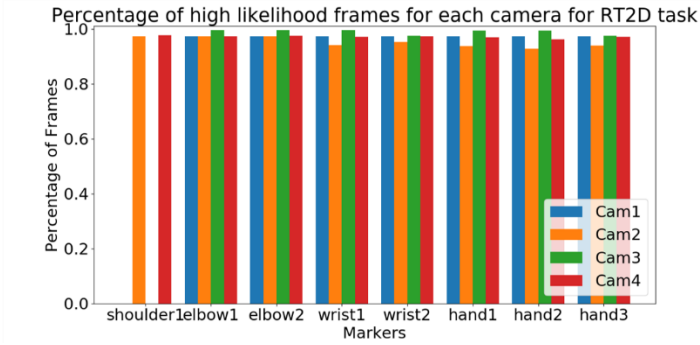
Besides knowing the model's performance, it is also important to check how well the cameras can track each landmark. To evaluate this, I used the likelihood hood parameter introduced by DLC. For RT3D task (Figure 16-A), the two lower cameras (2 and 4), which were meant to track the shoulder and elbow landmarks, had a great performance in doing so. The percentage of high likelihood landmarks (likelihood > 0.6) for these two cameras was 97%. Camera 4 also did well in tracking the wrist and hand landmarks with 90% high likelihood landmarks, providing yet another camera view for 3D reconstruction. The two upper cameras (1 and 3) also achieved a high percentage (93% and 91%) of high likelihood frames while tracking their designated landmarks, suggesting that all 4 cameras had a good position to not only track their intended landmarks, but also extra unintended landmarks, providing extra references in 3D reconstruction.

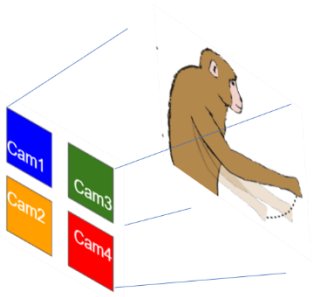
RT2D task (Figure 16-B), with its smaller workspace, was easier to record and 3D reconstruct. The percentage of high likelihood landmarks in all camera views were above 90%, except shoulder1 in cam1 and cam3. Most of the cameras (cam1, cam3, cam4) reached 98%, indicating that in RT2D task, all 4 cameras tracked the landmarks well on the non-tattooed monkey.

A (RT3D task)



B (RT2D task)





C

Figure 16: Percentage of high likelihood landmarks for each landmark in each camera for a RT3D task (A), and a RT2D task (B) on only the non-tattooed monkey. And a reference picture of the camera positions relative to the monkey (C).

3D reconstruction performance

To evaluate 3D reconstruction accuracy, 3D-2D reprojection error serves as a good benchmark. It is used to quantify how close the true landmark position in the camera space is to the estimated landmark position in the 3D space. I tested the reprojection error for both the non-tattooed monkey (monkey C) and the tattooed monkey (monkey H), on both RT3D and RT2D tasks. The “elbow”, “wrist” and “hand” datapoints mentioned in Figure 17 were calculated by averaging the reprojection error over both the elbow landmarks, both

the wrist landmarks, and all 3 hand landmarks. The overall reprojection error for both monkeys in both conditions is 3.7 ±2.5 pixels. The shoulder landmark has the least reprojection error with 1.4 ±1.7 pixels, while the elbow, wrist and hand landmarks have nearly the same reprojection error around 4.0 - 4.3 ±2.4 pixels.

Additionally, RT3D tasks generally have reprojection errors with larger distributions (4.1 +/- 3.0) compared to RT2D tasks (3.6 +/- 2.0). On the other hand, the reprojection error between Han (4.0 +/- 2.1) and Crackle (3.8 +/- 2.8) did not show much difference.

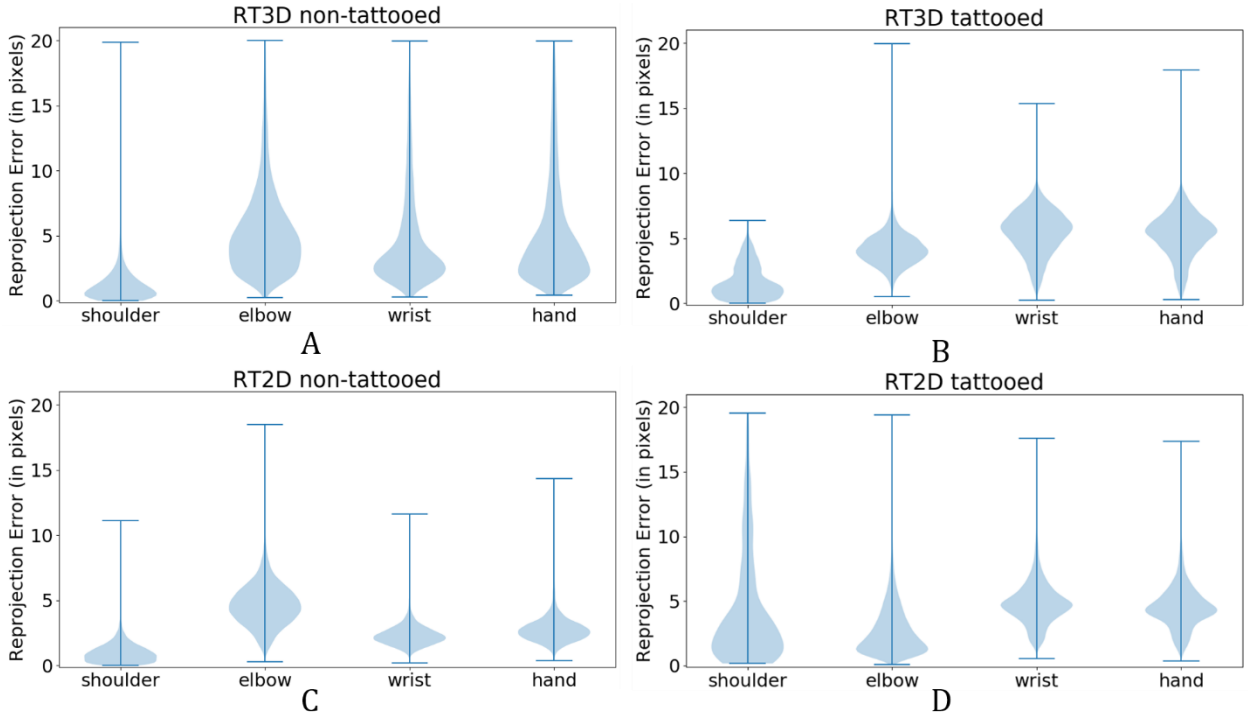


Figure 17: 3D to 2D reprojection error for a non-tattooed monkey (A, C) and a tattooed monkey (B, D), in RT3D and RT2D tasks.

Next, I looked into the kinematics of the monkey's movements in both tasks. Figure 18 shows the position distribution of hand2 in both RT2D task (blue points), and RT3D task (red points). As expected, the RT3D task allows the monkey's hand to move in a larger space compared to the RT2D task, which only allows the monkey to move its hand in a plane.

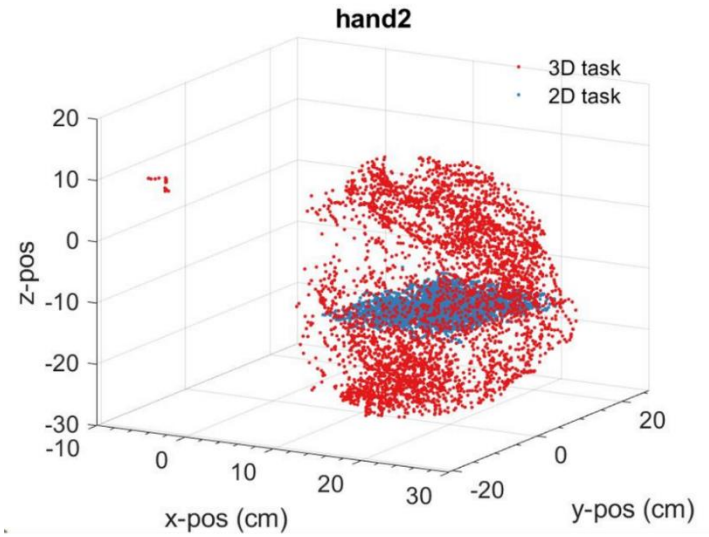


Figure 18: Hand2 landmark's position of the non-tattooed monkey in space comparing between RT2D task and RT3D task. The red points are the positions of the hand in RT3D task, and the blue points are the positions of the hand in RT2D task.

In addition to position, another kinematic feature that we care about is the speed of the arm. Figure 19 shows an example 1.5 second trajectory of the arm when reaching to the target in RT3D task (Figure 19-A) and RT2D task (Figure 19-B). The larger interval

between consecutive stick figures in the RT3D task than the RT2D task indicates that the monkey is moving its arm faster.

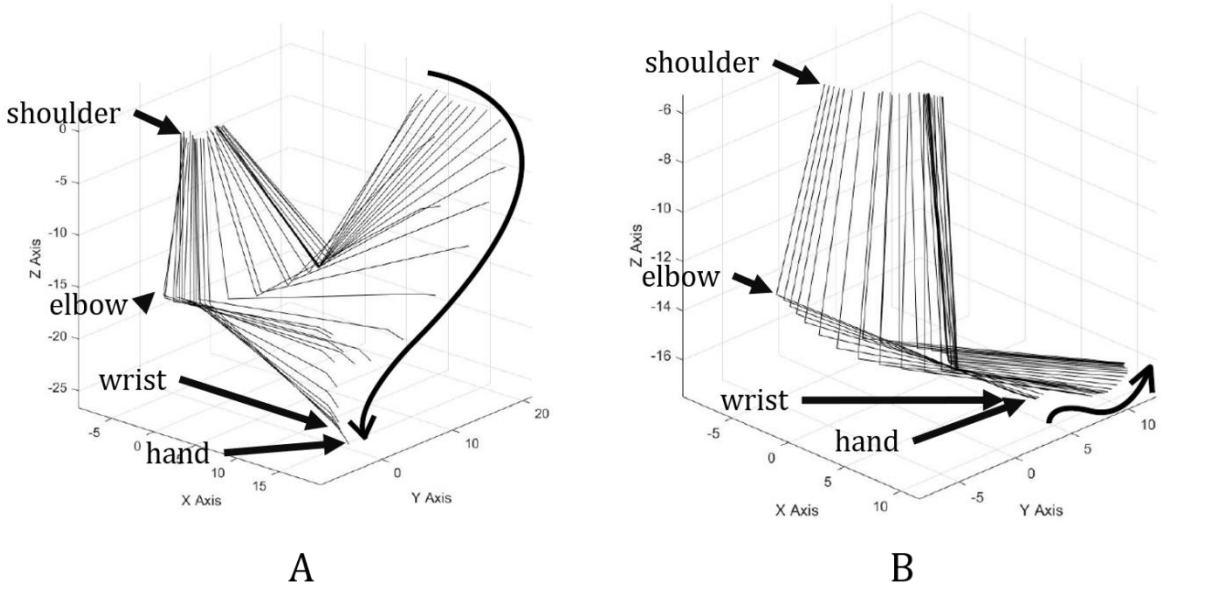


Figure 19: Example trajectory of arm movements in RT2D and RT3D task. (A) Example trajectory of landmarks on the arm during a reaching movement in RT3D task. (B) The same landmarks for RT2D task. The shoulder, elbow, wrist and hand landmarks for each frame are linked by thin black lines. 30 consecutive frames were chosen for both figures, with an interval of 0.05s.

Comparing the speed distribution of a hand landmark between a RT2D and a RT3D dataset shows that overall, monkey H moved its arm faster in RT3D task than RT2D task. In Figure 20, it can be seen that for RT2D task, the hand speed is mostly less than 0.6m/s. But for RT3D task, the peak of this speed distribution increased to 1.4m/s.

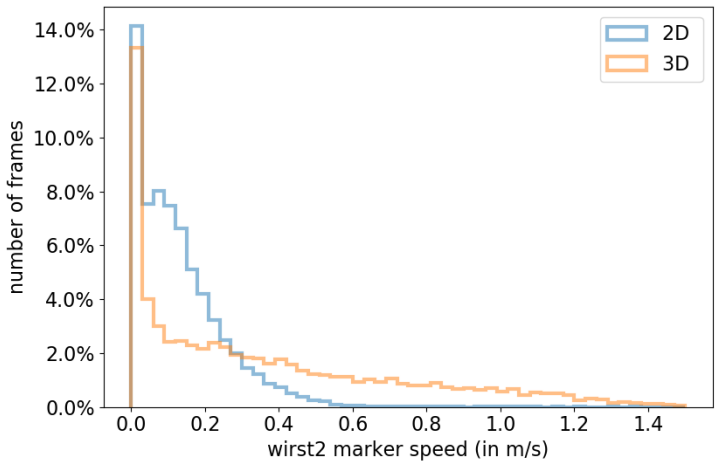


Figure 20: Hand speed data comparison between RT2D and RT3D task. X axis represents the speed of hand2 landmark in m/s, and Y axis represents the percentage of frames in the dataset with such speed.

Besides the overall comparison of the distribution of position and speed, it is important to know if RT3D, this new experiment paradigm, has the ability to decrease the correlation between elbow and hand landmarks. To test this, I took 4 experiment datasets, and, for each dataset, calculated the correlation between elbow and hand movement using the X and Y coordinates of velocity, and speed. In Figure 21, different symbols represent the correlation of different types of kinematic data, and different colors represent different datasets. Each line parallel to one of the axes represents the error bar of that experiment recording's kinematical data.

The results showed that even though for most of the datasets, the velocity on the X axis (Vx) are well decorrelated for the RT3D task, the velocity on the Y axis and the speed were not well decorrelated. The error bars are also huge for some of the points, indicating either

possible outliers in the dataset after 3D reconstruction, or huge correlation variations in different reaching movements. In general, this suggests that RT3D task probably did not meet our expectations in terms of decorrelating the elbow and hand movements.

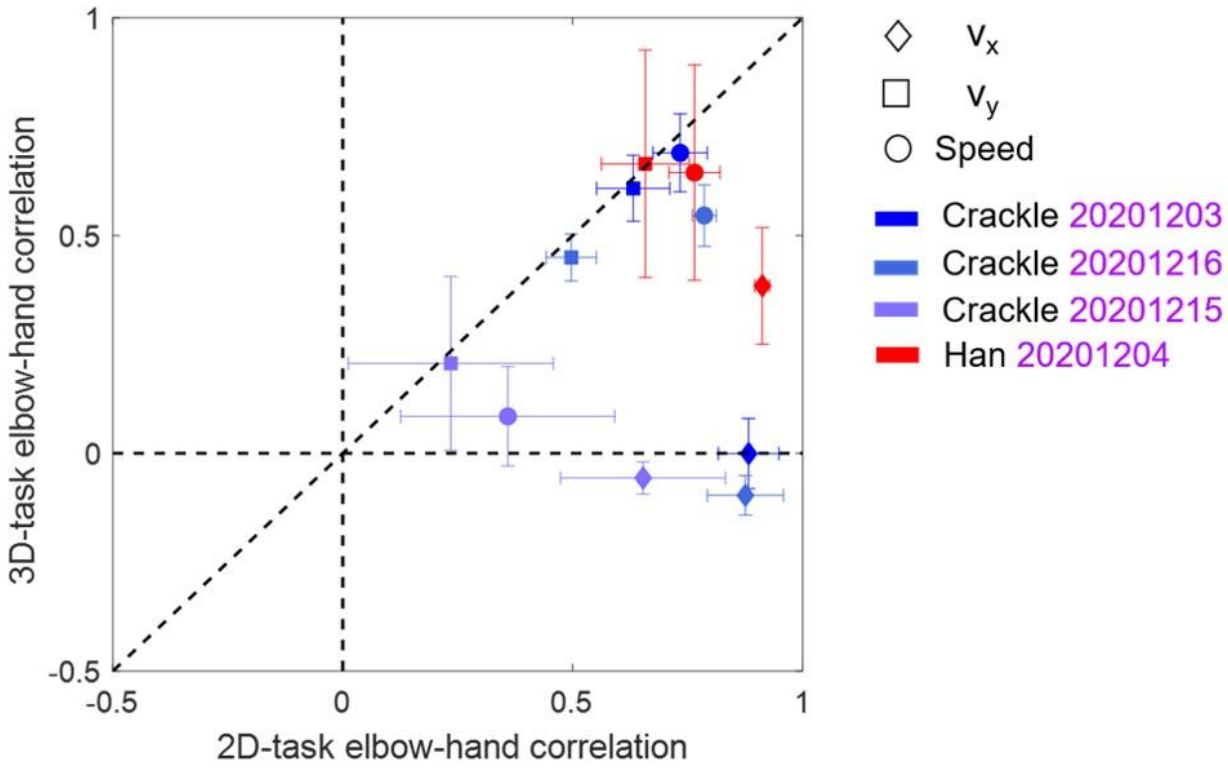


Figure 21: Comparisons of the hand-elbow correlations between RT2D and RT3D tasks over multiple datasets. The correlation values for the RT2D task are plotted along the x axis, and the correlation values for the RT3D task are plotted along the y axis. The symbol for each dot denotes the type of the kinematics variable on which the correlations are calculated, and the colors correspond to different datasets. The error bars show ± 1 standard deviation.

GLM's performance in predicting neural responses

I used GLMs to model the relationship between the upper limb kinematics derived from DLC and the neural responses simultaneously recorded in S1 area 2, from a tattooed monkey's RT3D task. Figure 22 shows examples of the predictions of neural firing rates, compared to the actual neural recordings. For each single neuron shown in Figure 22, I used both the hand-only and whole-arm models. The figure shows 3 representative neurons that have strong neural responses to reaching movements. Both models can predict whether this neuron would or would not respond to a specific movement, as shown in 1-2 seconds in column 2. Both models predicted the neural responses from all 3 neurons to this reaching activity. But sometimes, both models failed to predict the accurate firing rates of the neural response to a reaching movement in some of the neurons, for example, 1.5 – 2.5 seconds in column 1. In this reaching activity, both models predicted the neural responses for neuron 13 and 25 accurately but failed to do so for neuron 19. With that being said, both models seem to predict equally good and equally bad at the same time, with no obvious phenomenon that the whole-arm model predicts better than the hand-only model, given the same data. This also holds for the RT2D task as well. Also, there were roughly no difference of the pR^2 values between the two models in neuron 13 and neuron 25. There was some difference between the pR^2 values of the two models in neuron 19, but I'm neither sure about whether this difference is significant, nor confident that this difference is universal in the dataset. In general, the predicted firing rate using a whole-arm model was not more accurate than that using a hand-only model.

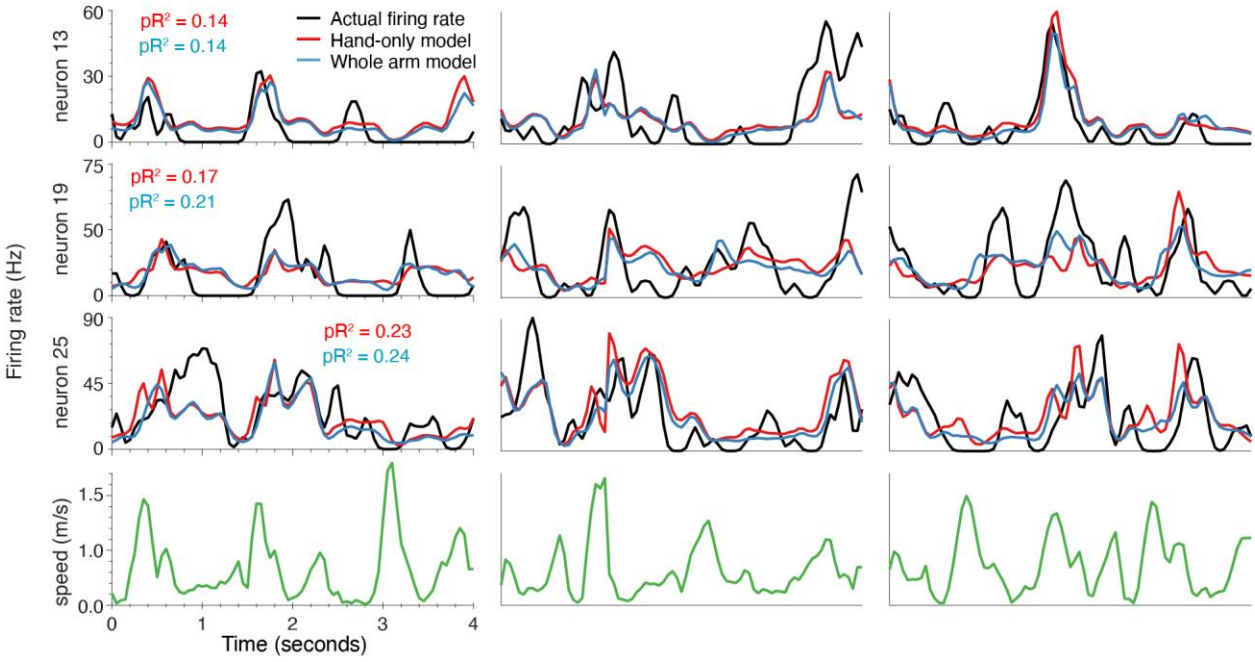


Figure 22: Actual and predicted neural responses during the RT3D task with monkey H. The major peaks in hand speed (~3 in each segment) represent reaches. Three 4-second representative sections of neural data from three neurons were chosen from the experiment recording. Black line shows the actually recorded neural firing rates, while red line shows the predicted neural responses using hand-only model, and blue line shows the predicted neural responses using whole-arm model.

Figure 23 shows a quantitative summary of the results for both tasks, using the data from monkey H. Each data point shows the pR^2 value of the hand-only model compared to that of the whole-arm model. Most of the pR^2 values of the neurons for both RT2D and RT3D tasks are slightly higher on the whole-arm model side compared to that of the hand-only model. Also, the pR^2 values for the neurons in both tasks adhere to the unity line, and I did not see that the pR^2 value of the neurons in RT3D task are closer to the whole-arm model side,

compared to the RT2D task. All this suggests that the whole-arm model in general only performs a little bit better than the hand only model for the majority of the neurons. It also suggests that the whole-arm model did not perform better than the hand-only model in the RT3D task compared to that in the RT2D task, as the hypothesis states.

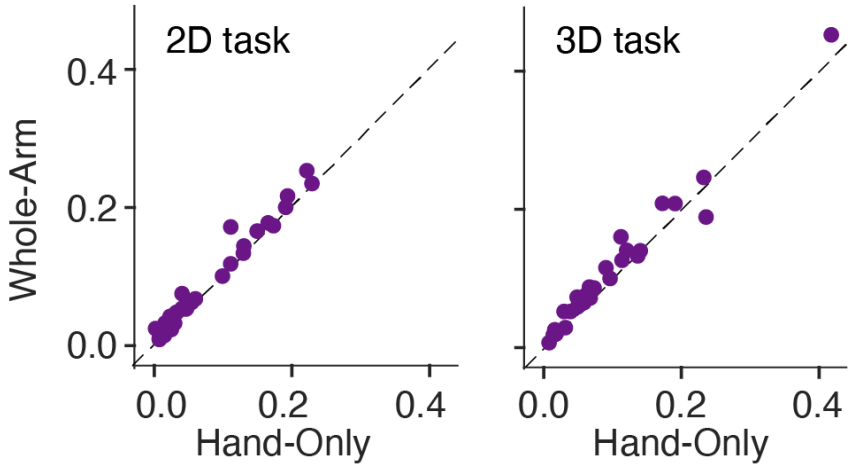


Figure 23: pR^2 comparison between whole-arm model and hand-only model in RT2D and RT3D task conditions. Each data point represents a single neuron.

Besides comparing the prediction performance between the two models, I also compared the same performance between the two tasks. In Figure 24, I showed the pR^2 difference between RT2D and RT3D tasks, for each pair of models using the same neuron. Since the majority of the points are very close to the unity line, it is possible that RT3D task, as an experiment paradigm, did not significantly outperform the original RT2D task in terms of model prediction performances, no matter which model being used.

Also, the hypothesis of this study would have been supported, if for each pair of models from the same neuron, the result of the whole-arm model went above the unity line, while the result of the hand-only model did not. Showing that would suggest that whole-arm model obtained more kinematical information in the RT3D task than the RT2D task, compared to the hand-only model. But since this was not observed in the figure, it would suggest that the whole-arm model probably has not learned anything new compared to the hand-only model using the RT3D task.

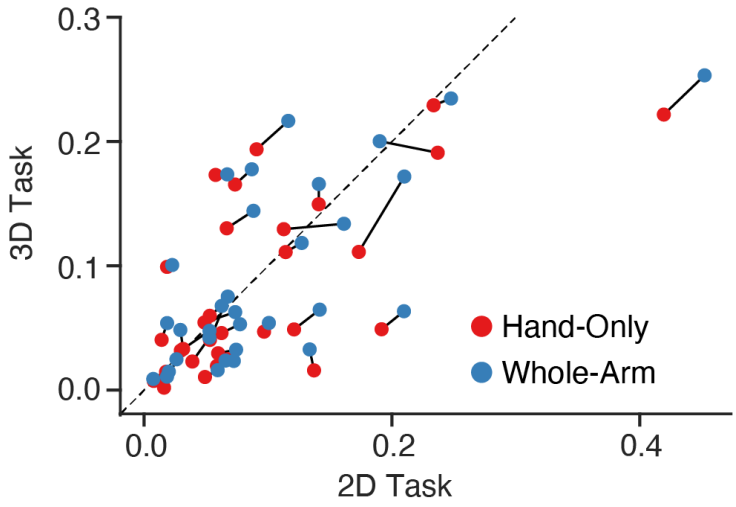


Figure 24: pR^2 comparison between RT2D and RT3D tasks. Each red dot represents the GLM hand-only model prediction performance, based on one neuron's data, compared between RT2D and RT3D tasks. Each blue dot represents the same thing, but for a whole-arm model. The black lines linking between each pair of red and blue dots mean that the dots represent the same neuron's performance from different models.

Discussion

Summary

In this study, I implemented and evaluated a recording and data processing pipeline to do markerless motion tracking of monkeys reaching in the 3D space. I optimized the pipeline so that with 4 cameras, all the arm kinematic information can be recorded in most cases with minimal problems. Second, with the pipeline mentioned above, I investigated how S1 area 2 neurons encode upper limb kinematics in monkeys with RT3D task, a new experiment paradigm. From the results, I found that first, RT3D task did not decorrelate the kinematic relationships between elbow and hand as much as I expected. Second, with the RT3D task, I did not see an increase of the whole-arm model's performance compared to the hand-only model, comparing to that in the RT2D task. Overall, these results suggest that either the current design of RT3D task could not adequately decorrelate the elbow and hand kinematics during reaching, or that the arm representation area of area2 in S1 does not contain much elbow kinematics for the whole-arm GLM model to improve its performance, compared to the hand-only model.

Factors influencing DLC performances

Several factors played important roles that still affect DLC's performance, even with an optimized video. The main factors described here are tattoos, labeling, training set size, training iterations, and model's ability to generalize across datasets, within and across monkeys.

The size of the training dataset and the number of training iterations are two of the most important factors that influence DLC's tracking performance. DLC's study tested several factors influencing tracking performance, including training iterations, number of training images, and model estimation error compared to human variability (Mathis et al., 2018). But since DLC's data is not solely based on monkey, it is necessary to re-test these factors in our environment. DLC's paper reported tracking performances based on an odor-guided navigation task on mice. The authors tracked the snout, ears and tail base of a mouse. They extracted and labeled 1080 frames across 2 cameras, 7 different mice, and used these frames to train varying models to compare the performances (Mathis et al., 2018). The test RMSE is calculated in terms of pixels. Results indicated that the model performance plateaued at 400k iterations with 864 frames, and 600k iterations with 520 frames. For both cases, the final plateaued RMSE was around 3.5 pixels. But in our test, the models for both the non-tattooed monkey and the tattooed monkey plateaued at 4 and 5 pixels, a higher RMSE than 3.5, with similar number of frames. This suggests that for different experiment environments and experiment paradigms, the number of training frames and iterations should be changed accordingly to achieve maximum performance.

Even though DLC can properly track landmarks on animals' skin without tattoos or reflective markers, having them on the skin still improves the DLC tracking performance. The average model prediction error for a non-tattooed monkey can be one pixel higher than a tattooed monkey. This suggests that if one wants to achieve maximum tracking accuracy anyway using DLC, adding a tattoo onto the tracking point might be one of the choices to decrease model prediction error.

Another important factor that might influence DLC's performance is the error in labeling by a single labeler, and by multiple labelers. For the error source from a single labeler, the main problem is that it is obviously impossible to label each frame accurately onto the exact position of the skeletal anatomical positions based on pictures of the skin, so in any case, the landmarks I labeled are estimates of the skeletal anatomical positions, based on fur colors and muscle shapes. This labeling error will increase if multiple labelers haven't learned the references for the landmark position well enough, This would result in a noisy input to DLC, causing bad training results. To reduce this noise, a complete guide to label the experiment animals in all positions would likely be of help.

The monkey's movements could also influence the tracking performance by getting the landmark obscured. The tracking performance of the hand landmarks, as we can see from the experiment results, no matter the monkey is tattooed or not, could not outperform the other landmarks in RT3D task, but could do so in RT2D task. This is likely because the monkey can move its hand in more directions in the RT3D task, thus sometimes obscuring the landmarks on the hand. Since DLC does not generalize the positions of the landmarks

through the body (generalizing from the back of the hand to the front of the hand, representing the same joint), using only four cameras to cover the range of the hand's activity is tough. The landmarks will be obscured if the monkey decides to flip its arm or twist its arm into an extreme angle, which might better decorrelate the elbow and the hand, but might also lose the hand's position. To cover the extreme angles of the hand, extra cameras from extreme angles should be useful.

Another factor influencing DLC's performance is its ability to generalize across datasets and monkeys. When processing the dataset for the non-tattooed monkey, if I feed in the dataset doing the same task on day N+1 to the model trained using only the data from day N, the tracking performance of some of the markers will go down, with the appearance of the predicted landmark vibrates around the actual landmark position, even after filtering. This suggests that DLC still thinks that the landmark position it is predicting is likely to be true, but the actual prediction results come to be unstable. This will probably affect the ongoing data processing accuracy if a kinematical model will be constructed based on this predicting result.

Area 2 neurons encode upper limb kinematics in a similar way in RT2D and RT3D tasks

The final results, the difference between the pR^2 values of the whole-arm model and the hand-only model using RT3D task, did not meet my expectations. I expected to see that the difference between the two models in RT3D task is higher than that in the RT2D task, so that it would support the idea that area 2 neurons did actually encode whole-arm kinematics. Instead, the current results were similar to the results from the previous study (Chowdhury et al., 2019), showing that RT3D task did not improve the performance of the whole-arm model compared to that of the hand-only model. This might suggest that area 2 neurons did encode upper limb kinematics in a similar way in RT2D and RT3D tasks.

By comparing the workspace size and landmark speed between the two tasks, the RT3D task did significantly increase the workspace and allow the monkey to reach faster. But both changes did not significantly decorrelate the kinematic relationships between elbow and hand. Results showed that even though for all the datasets, the mean correlation of all types of kinematic data (velocity on X and Y axis, speed) of the RT3D task were smaller than that of the RT2D task, most of the differences were not significant. One possible reason that caused this small correlation difference is the monkey's higher number of degrees of freedom (DOF) when doing the RT3D task, compared to that of the RT2D task. As we have previously discussed, the landmark on the monkey's hand might be occluded if the monkey turns its hand to an extreme position in the RT3D task. To keep the recording

quality, it is necessary to keep the monkey's back of the hand facing the camera as much as possible. This cuts off some of the DOF, and possibly restricting the decorrelating movements the monkey is making. Another possible reason is that by restricting the monkey on the chair, it limits some of the more decorrelating movements it can make in a non-restricted environment. This would also possibly increase the average correlation in the dataset. A way to reduce the correlation between joints would be to design experiment paradigms that allow the monkey to move even more freely in the workspace. One example would be to provide a cage that the monkey can freely move in and let the monkey reach to targets in or around the cage. This would prevent restrictions of any other body parts of the monkey and provide a dataset with even more unconstrained reaching movements.

The previous paragraph assumed that the area 2 neurons actually encode different information from RT2D and RT3D tasks, meaning that there should be a difference between the two models' performance in between the two tasks. Another reason why we had results that did not go along with our expectations is that, in both RT2D and RT3D tasks, area 2 neurons actually encode the kinematic data in a similar fashion. By comparing the goodness-of-fit results between the two models in both RT2D and RT3D tasks, I showed that the difference between the two models is not significant in the RT3D task, compared to that using the RT2D task. This does not fit our hypothesis, that the whole-arm model would perform even better relative to the hand-only model in RT3D task compared to the RT2D task, since RT3D task would potentially include more kinematical information for the whole-arm model by decorrelating the kinematic relationships between elbow and hand.

If we assume that RT3D task does include more kinematic information, the difference between the whole-arm model and the hand-only model in RT3D task should be larger compared to that in the RT2D task, as the hypothesis suggested. But since this is not seen in the results, a likely hypothesis would be that there is not extra neural information in the RT3D task for the whole-arm model to learn about. In other words, area 2 neurons encode similar kinematic information in both RT2D and RT3D tasks.

This similarity might then cause both models to learn to use very similar information when predicting the neural responses, thus resulting in similar neural prediction performances. A possible reason why this happened is that area 2 neurons are on the more "high-level" side of this encoding pathway. In other words, area 2 neurons might strongly encode the hand kinematics, but do not care about the kinematic information from other joints, such as elbow.

Future work

The current two main problems are that, can RT3D task successfully decorrelate between elbow and hand kinematics, and can whole-arm model actually learn more from RT3D task than RT2D task, compared to hand-only model?

Figure 21 shows that only the correlation of Vx dropped for the RT3D task, while Vy and speed still remained the same. This suggests that maybe some of the experiment methods that we used in the RT3D task worked to decorrelate the joints but did not get applied onto the other axes, or maybe there was noise in Vx. In the future, I will either let the monkey do different reaching tasks in the chair with movements that better decorrelate the upper limb kinematics, or let the monkey do more unconstrained reaches in the cage. Both of these experiment paradigms might serve to better decorrelate the joints.

In this study, I only choose to use two firing rate encoding models that were used in the previous study, the hand-only model and the whole-arm model. The hand-only model serves as a baseline, and the whole-arm model serves as the best result from the previous study. Other models, like hand kinematics + force model, egocentric model, joint kinematics model, and muscle kinematics model, are all not included in my data analysis. It is possible to check if area 2 encodes any other kinematical information from the upper limbs or not, using different firing rate encoding models.

Another way to move forward would be to try other models rather than GLM, to see if that would increase the encoding performances, and why that might be the case. One of the choices would be to use an Artificial Neural Network (ANN) to dig into the encoding relationships between kinematical data from the arm and the neural data (Lucas et al., 2019). I might get a good prediction result from this model, but the complexity of this model would make it hard to interpret these results. Ultimately, using a new and more complicated model might not help understanding the brain better.

Conclusion

The goal to conduct this study was to improve the understanding of how area 2 neurons encode upper limb kinematics in unconstrained environment. I began by constructing a new video recording and data processing pipeline, aiming to replace the previous pipeline for a more complicated behavior task, the RT3D task. The construction result was successful, and I can record and analyze kinematic data with this pipeline. I then proceed to analyze the GLM's performance in predicting neural responses, using the whole-arm model and the hand-only model, in both RT2D and RT3D tasks. Results suggested that area 2 neurons encode upper limb kinematics in a similar way in RT2D and RT3D tasks.

References

Ager, A. L., Borms, D., Deschepper, L., Dhooghe, R., Dijkhuis, J., Roy, J. S., & Cools, A. (2020). Proprioception: How is it affected by shoulder pain? A systematic review. *Journal of Hand Therapy*, *33*(4), 507–516.

Bardsley, D. (n.d.). 3D Reconstruction Using the Direct Linear Transform with a Gabor Wavelet Based Correspondence Measure Technical Report. *Bardsley.Org.Uk*, 1–11.

Chowdhury, R. H., Glaser, J. I., & Miller, L. E. (2019). Area 2 of primary somatosensory cortex encodes kinematics of the whole arm. *BioRxiv*, 1–33.

Deng, J., Dong, W., Socher, R., Li, L.-J., Kai Li, & Li Fei-Fei. (2010, March 1). *ImageNet: A large-scale hierarchical image database*. 248–255.

Ghez, C., & Sainburg, R. (1995). Proprioceptive control of interjoint coordination. *Canadian Journal of Physiology and Pharmacology*, *73*(2), 273–284.

Gordon, J., Ghilardi, M. F., & Ghez, C. (1995). Impairments of reaching movements in patients without proprioception. I. Spatial errors. *Journal of Neurophysiology*, *73*(1), 347–360.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, *2016-December*, 770–778.

London, B. M., & Miller, L. E. (2012). Responses of somatosensory area 2 neurons to actively and passively generated limb movements. *J Neurophysiol*, *109*, 1505–1513.

Lucas, A., Tomlinson, T., Rohani, N., Chowdhury, R., Solla, S. A., Katsaggelos, A. K., & Miller, L. E. (2019). Neural networks for modeling neural spiking in S1 cortex. *Frontiers in Systems Neuroscience*, *13*(March), 1–10.

Mathis, A., Mamidanna, P., Cury, K. M., Abe, T., Murthy, V. N., Mathis, M. W., & Bethge, M. (2018). DeepLabCut: markerless pose estimation of user-defined body parts with deep learning. *Nature Neuroscience*, *21*(9), 1281–1289.

Prud'homme, M. J. L., & Kalaska, J. F. (1994). Proprioceptive activity in primate primary somatosensory cortex during active arm reaching movements. *Journal of Neurophysiology*, *72*(5), 2280–2301.

Sainburg, R. L., Poizner, H., & Ghez, C. (1993). Loss of proprioception produces deficits in interjoint coordination. *Journal of Neurophysiology*, *70*(5), 2136–2147.

Sanes, J. N., Mauritz, K. H., Evarts, E. V., Dalakas, M. C., & Chu, A. (1984). Motor deficits in patients with large-fiber sensory neuropathy. *Proceedings of the National Academy of Sciences of the United States of America*, *81*(3 I), 979–982.

Weber, D. J., London, B. M., Hokanson, J. A., Ayers, C. A., Gaunt, R. A., Torres, R. R., … Miller, L. E. (2011). Limb-state information encoded by peripheral and central somatosensory neurons: Implications for an afferent interface. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, *19*(5), 501–513.