GEL-7072 Bioinstrumentation & Biomedical Microsystems

Neural Decoding from Rodent's Primary Motor Cortex

Ayoub.E *Pierre.N*

Official Project Report
Presented to Mr. Benoit Gosselin

Faculty of Science and Engineering
Laval University
Fall 2022

<u>Sections</u>

1) Context & Problematic

Numerous problems still exist in order to accurately map neural signals in the motor cortex to the corresponding behaviours of a free-behaving subject. Achieving this feat could represent access to new kinds of motor and communication prosthetics that would benefit society as a whole. Although the recent utilization of deep learning techniques as non-linear approximator for the mapping between spikes and motor actions has shown great potential in increasing the decoder's accuracy, getting the right amount and diversity of data necessary to train those models is still an intense challenge as decoders are still resource-intensive (time & personnel) to create. Therefore, finding a way to build training samples (Neural firing Patterns -- Corresponding Kinematics) using free-behaving subjects could be a way to reduce the amount of resource necessary to train accurate neural decoders.

For this purpose, using automated label collection systems in conjunction with devices recording large amounts of neural signals discretely while a subject may be performing daily activities may represent an innovation in the engineering of Brain-Computer Interfaces. Additionally, the usage of techniques for synthetic spikes-kinematics data generation that could multiply the size of decoder's datasets by 10, 20, or 30 times could strongly enhance the density of the function space, hence avoiding error in the decoding process due to poor training in a specific space of the data manifold. On that account, in order to maximise the effect of a scaling in the amount of data and avoid large latency in the inference process, designers of neural decoders may need to consider alternatives to the typical recurrent LSTM models and opt for the usage of recent advances in high-capacity models like the Transformers, which are well suited for modeling any type of sequence-to-sequence problems.

2) Objectives

With the problematic above in mind, the objective of our research project can be resumed as investigating the possibility of decoding from the motor cortex of a free-behaving rodent the corresponding behaviours using a neural decoder build with two core concepts. First, we should design and implement a **label collection system (LCS)** for collecting the behaviours and kinematics from rodents via video-capture, while recording their motor cortex activity using implanted micro-electrodes. Second, after processing the data and building our datasets, we should try to train a neural decoder that make usage of the **Sequence-to-Sequence Deep Learning architecture** and make comparison between alternatives. Although we are well aware that we are experimenting with a simplification of the real task of producing accurate neural decoders with multi-task generalization in free-behaving context, this experiment is conducted only for demonstration purpose. We are of course thankful to Dr. Éthier and the personnel of his laboratory at the CERVO research center for their assistance in this project.

3) Literature Review

3.1 -- Hardware for High Density Neural Recording

Although our project does not involve the specific design and testing of any hardware component due to the already available recording technology present at Dr. Éthier's laboratory, we nonetheless insisted on reviewing the state-of-the-art in high density neural recording systems due to its importance in our course content but also due to its critical role in the engineering of almost any modern bi-directional neural interfaces. Hence, two systems will be reviewed here, which are: the Argo recording system by Paradromics [*Sahasrabuddhe & al*, 2020] and the electrophysiological system from Neuralink [*Musk and Neuralink*, 2019]. In the following sections, those will be **analyzed when it comes to the 3 following phases**: signal acquisition, amplification & filtering, and finally digitization & transmission.

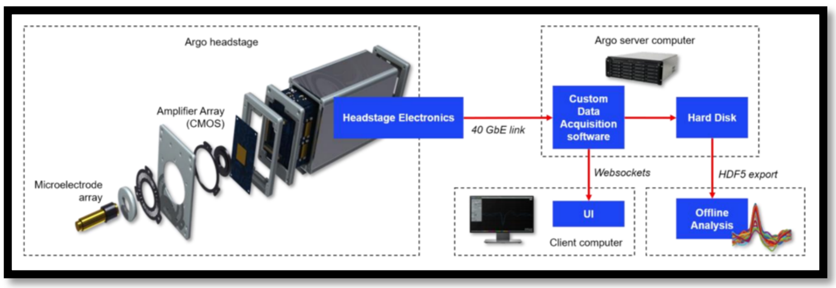


Figure 1: *Diagram of the Argo System. The focus of this review revolve around the Argo headstage, which includes the recording electrodes, the CMOS amplifier array, and electronics to digitize, packetize, and transmit signals over an optical data link.*

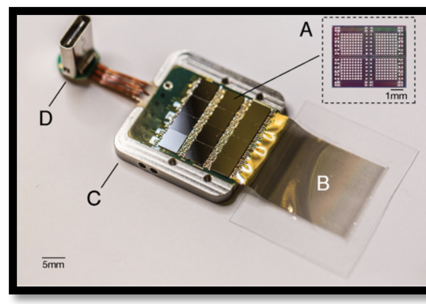


Figure 2*: Neuralink packaged sensor device comprised of 12 individual neural processing ASIC seen in (A) capable of processing 256 channels of data each. In (B), we can see the Polymer threads on parylene-c substrate. And in (C) and (D), we can see the Titanium enclosure and the USB-C connector for power & data.*

A) Signal Acquisition

The **Argo recording system** uses an array of platinum-iridium (90% Pt/10% Ir) microwire electrodes that are connected to an active CMOS electronic system for readout and stimulation. Microwire electrodes can be understood as a conductive metal wire core insulated by a solution-resistant dielectric (ex: polymer or ceramic). Those were selected as they are proven, highly robust, and suitable for chronic recording applications. Also, the material core was selected for his strong biocompatibility and recording/stimulation performance in vivo. For intracortical arrays specifically, the ends of the microwire electrodes are electro sharpened (See figure 1 B), the final tip diameter is smaller than 200 nm, and arrays are coated with a robust 200-300 nm Atomic Layer-Deposited (ALD) alumina coating to provide a high-quality insulation layer that is then selectively de-insulated to define the length of the recording site at the wire tip (See Figure 1 A). Finally, a macroscopic view of the electrode array can be seen on figure 1 C.

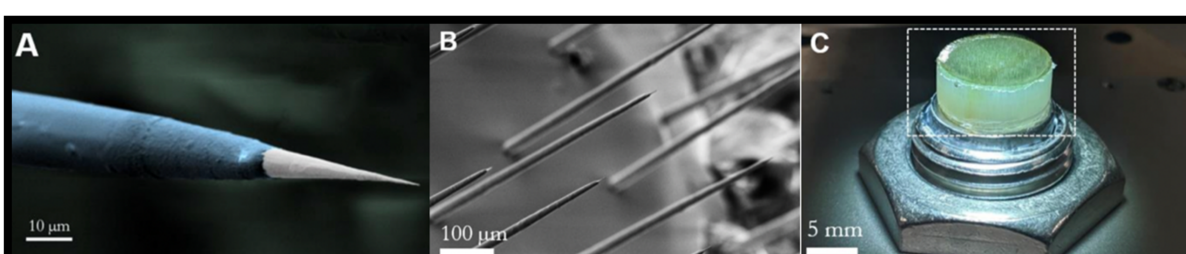


Figure 3: *(A) Alumina coating acting as an insulation layer in blue while the white region represents the recording site. (B) View of the microwire electrodes electrosharpened ends (C) Macroscopic view of the electrode array*

When it comes to the **Neuralink System**, they took a different path and opted for the use of **ultra-fine polymer probes** in order to allow a degree of flexibility that can minimize the micromotion between the probe and brain tissue, hence allowing better long-term integration properties for their system. Their thread design allowed them to scale the number of electrodes per thread, reaching 32 independent electrodes on a single thread in the proposed design. In figure 2 A and B, we can see two different configurations from numerous tested ones: the (A) “Linear Edge” probes with 32 electrode contacts spaced by 50 µm, and the (B) “Tree” probes with 32 electrode contacts spaced by 75 µm. The thickness of a thread is kept between 4 to 6 µm and its length is only 20 mm. Each thread interfaces with the integrated chip at a “sensor” location that enable signal amplification and acquisition. And as an array may possess 48 or 96 threads in total, the number of electrode contacts can be scaled to 3,072.

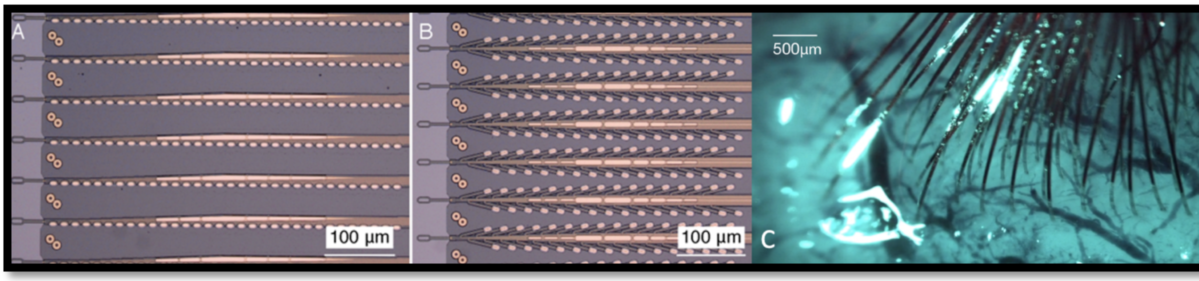


Figure 4: *(A) “Linear Edge” probes design with 32 electrode contacts spaced by 50 µm. (B) “Tree” probes design with 32 electrode contacts spaced by 75 µm. (C) An example showing the cortical surface of a rodent subject with implanted threads.*

Transferring to the materials and electrical properties of the system, the main substrate and dielectric used for their probes is **polyimide**, which encapsulates a **gold thin film trace**. To lower the impedance and increase the effective charge-carrying capacity of their interface as their individual gold electrode sites were made with small surface areas, they used surface modifications via treatment with either electrically conductive polymer poly-ethylenedioxythiophene doped with polystyrene sulfonate **(PEDOT: PSS)** or the traditional iridium oxide **(IrOx)**. According to their analysis, the lower impedance of PEDOT: PSS is promising, but they expressed the trade-off made due to the known stability and biocompatibility of Ir-Ox.

Although the use of microfabrication techniques applied to flexible thin-films probes allowed to drive down the patterned features down to the size of an electron beam with metal films at sub-micron resolution, the quest for smaller probes of this kind is facing physical challenges. For example, as the wire get smaller, it increases the resistance, hence making it more difficult to separate the signals from the noise. And like mentioned by the team responsible for the Argo System, flexible thin-films stacks (e.g., Polyimide + metal) suffers from drawbacks that according to them, made the use of microwire-CMOS arrays more attractive when it comes to immediate scaling. Indeed, they stated that polymer-substrate probes are said to suffer from **cracking** and **delamination between insulation and electrode**, which can limit their effectiveness when compared to other approaches.

B) Amplification & Filtering

The CMOS Sensor of the Argo system is an application-specific integrated circuit (ASIC) designed to amplify and filter neural signals coming from the high-density microwire array. It was designed with great flexibility as it can be bonded to microwire arrays of varying electrode specifications via the mechanical pressing of the array on the sensor. When it comes to the fabrication, they used the 180 nm CMOS process technology node to produce a **pixel array of 256 x 256 pixels**, each pixel with dimension 50 µm x 50 µm. Hence, the active area of the readout array is considered to be around 12.8 mm x 12.8 mm, bringing the total dimensions of the ASIC to 14.5 mm x 16 mm if we include the peripheral circuit elements. Each pixel possesses a top metal pad (40 µm x 40 µm) which is used for the bonding with the microwire electrode. More specifically, this top metal pad is **AC-coupled to a low-noise amplifier (LNA) chain**, which is composed of **three main blocks** that we will review below:

- **Input Amplifiers** (acting also as a tunable high-pass filter): Each amplifier has a tunable input biasing circuit which in conjunction with an AC coupling capacitor forms a tunable high pass-filter. This block aims to remove DC offsets, drifts and slowly varying out-of-band high amplitude signal and to provide the system to operate with two modes. In one mode, setting the corner frequency to 18 Hz enables both neuron spiking activity and LFP to be recorded, since the latter has most of its integrated power below 100 Hz. In the second mode, LFP signals can be rejected by increasing the corner to higher frequencies. This allows for spiking signals to retain the entire dynamic range.

- **Antialiasing Low-pass Filter**: This block aims to keep the integrated noise floor as low as possible while simultaneously providing antialiasing. The amplifier chain is followed by a third order low-pass filter with a tunable corner frequency between 8 kHz and 50 kHz.

- **Output Column Buffer:** This aims to provide isolation between the pixel output and the column line used to multiplex the pixels in a single column. That is, when one of the pixels in the column is being read out the other pixels have their outputs disconnected from the line to avoid overloading. This reduces crosstalk between pixels and ensures that each pixel can read out signals from unique electrodes.

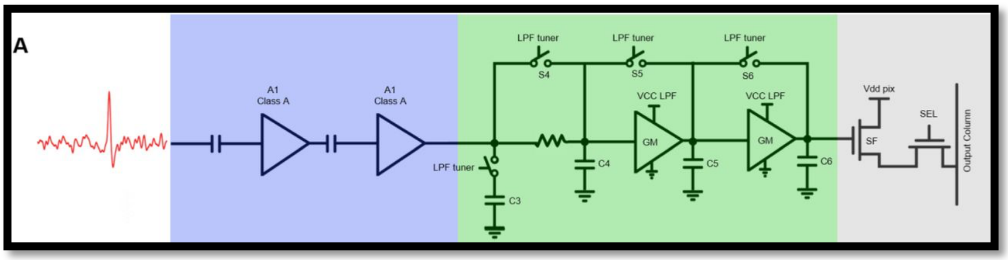


Figure 5: *(A) The input signal is AC-coupled into two source followers biased in the Class A configuration (A1, A2, shown in blue), which together form the front-end low noise amplifier (LNA) chain. Each stage has a gain of 10 V/V. The third stage (green) is a third-order tunable low-pass filter that serves as an anti-aliasing filter. The final stages (grey) are for pixel selection to read out the stored value. [1]*

Switching to the recording stack from Neuralink, which must first amplify small neural signals (<10 µVRMS) while rejecting out-of-band noise, then sample and digitize the amplified signals, and followed by the streaming of the results for real-time processing—all that using minimal power and size. The characteristics of the amplification are given in the table to the right.

Type	Value
Number of channels per device	256
Gain (dB)	42.9-59.4
Bandwidth (kHz)	3-27
Input-referred noise (3 Hz-10 kHz) (µVRMS)	5.9
Maximum differential input range (mVPP)	7.2

C) ADC & Transmission

The Argo System Electronics consist of two custom printed circuit boards (PCBs): a frontend board that is designed to house the wire-bonded CMOS sensor and support electronics, and the main board, which is designed to digitize analog signals and deliver them to a server and is characterized by two 16-channel high-speed ADCs in addition to a Field-Programmable Gate Array (FPGA). A main system overview can be gained by looking at the figure below:

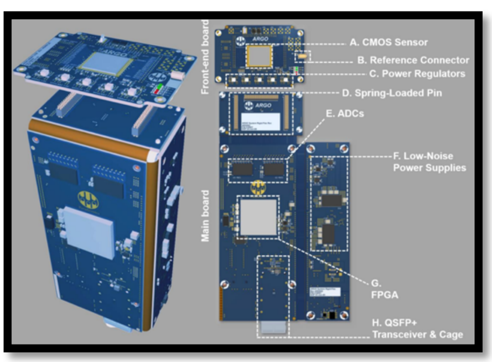


Figure 6: System Electronics

PCB 1: Front-end board
(A) CMOS sensor
(B) Reference connector
(C) Power regulators for the sensor.
(D) Three spring-loaded connectors for main board linkage

PCB 2: Main board
(E) Analog-to-digital converters (ADCs)
(F) Low-noise power supplies
(G) FPGA
(H) Quad Small Formfactor Pluggable (QSFP+) transceiver and connector

The ADCs operate at a quite high sampling frequency (~32 kHz) which allows further protection from aliasing artifacts. The ADC's input signal range of 2 Vpp translates to an input signal range of ∼2.5 mVpp in the CMOS amplifiers, given the 800 V/V gain. Concerning the Neuralink system, the sampling rate is lower (~19kHz). We also found a third system called 3D Bundle that was interesting, but we did not have space to add more content.

	Technology	Channels / ASIC	Sampling Rate	Resolution	Material	Total Signal Gain
The Argo **(1 ASIC)**	Microwires	256 x 256 (65,536)	32 kHz	12 bits	Platinum-Iridium	800 V/V
Neuralink System **(12 ASICs)**	Polymer Probes	256	18-19 kHz	10 bit	Polyimide/Gold + Surface Modification.	42.9 - 59.4 dB

3.2 -- Label Collection Systems (LCS)

Label collection Systems are fundamental to any neural decoder trying to produce a mapping between neural signals and motor activity, hence the necessity of a good system that collects the target labels in the training phase for producing an accurate decoder. If the experiment is trying to build a **kinetics-based decoder**, the target labels need to be the muscular activity of the free-behaving subject across time. Alternatively, a **kinematics-based decoder** will seek as target labels the dynamics of the different joints from the subject. The higher the number of joints, the higher the complexity of the mapping, which requires more resources, which means more input channels from the brain, more data to train on, and higher capacity models.

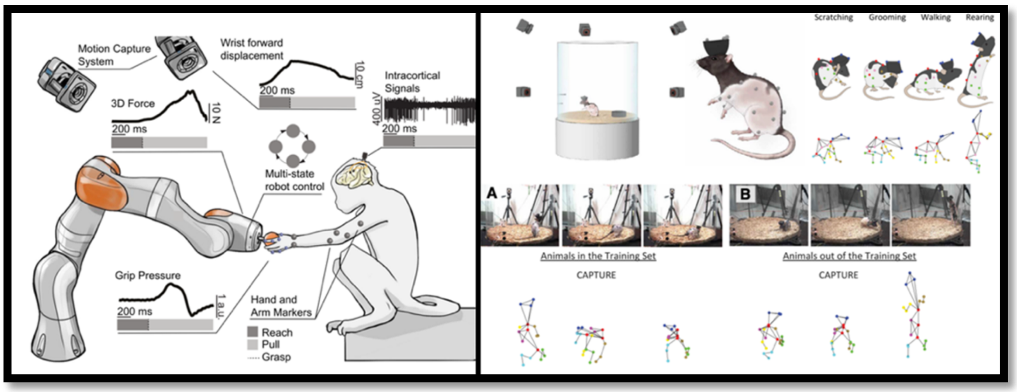


***Figure 7: Right**: LCS adapted to the extraction of 3D kinematics and interaction forces for monkeys. (B Barra et al 2020). **Left**: LCS for Rodent kinematics extraction in 3D by ([5] J. D. Marshall et al 2021).*

Traditionally, video-capture systems are the traditional way to go to extract the target labels in the training phase of a kinematics-based decoder. For example, a good LCS adapted to the extraction of 3D kinematics and interaction forces for monkey subjects is the one by ([4] B Barra et al 2020) seen on the left in figure 5, where neural recording from the sensorimotor areas can be mapped to multiple variables like the ones extracted by their motion capture system tracking the kinematics of the monkey's arm and fingers using reflective markers. Another LCS that is more adapted to our case due to its applicability to Rodent kinematics extraction in 3D is the CAPTURE System by ([5] J. D. Marshall et al 2021) seen on the right in figure 5, which consist of a motion capture system and some deep learning techniques designed for very high tracking accuracy that seems to outperform to a large extent traditional method used in the field like Deeplabcut, a popular framework for 2D and 3D pose estimation on rat and other animals.

3.3 -- Architecture for Free-behaving Neural Decoders

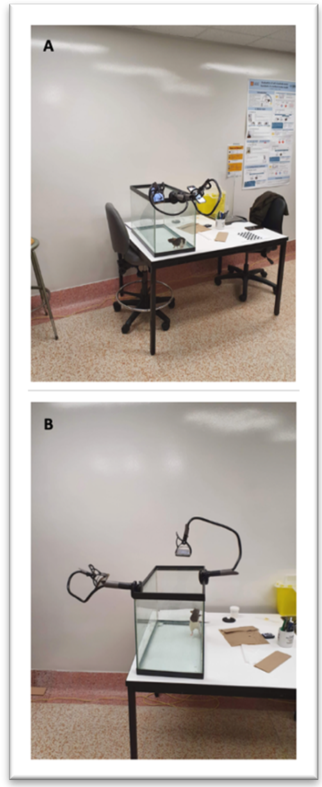
For the architecture of our neural decoder, we kept two possibilities in mind depending on the outcome and uncertainty of the experiments. First, we could perform continuous decoding of the joint's dynamics moment-by-moment in 3D, hence mapping the neural activity into continuous-valued movement trajectories for each specific joint that solve a regression task. Otherwise, if the experimentation and the collected neural data contained a lot of noise and not enough clear information signal for decoding trajectories, we would instead classify different distinct behaviours using the spiking activity extracted from the motor cortex. The following research [6] is a good example of relating neural activity to freely moving behavior through a classification framework while [7] and [8] demonstrated a mapping from neural activity to 3D joint's dynamics through a regression framework. Also, there is a possibility of combining both regression and classification by first decoding the continuous joint's trajectory, and then classifying the generated results to the corresponding classes (behaviours in this case). Recently, Neuralink [9] demonstrated their preference for this approach on Monkeys while testing their communication prostheses. Finally, in both a regression or classification task, we decided that our mapping architectures will be using neural networks like LSTMs or Transformers due to their large non-linear approximative power.

4) Experimental Setup

4.1 -- Label Collection System

- Requirements:

The requirements for the LCS system were in the first place the **kinematics extraction** of the joint's dynamics via 3D Pose estimation of the rat at each time step. Secondly, the manual labeling of the different **behaviour types** for a classifier would be performed via the analysis of the captured frames.

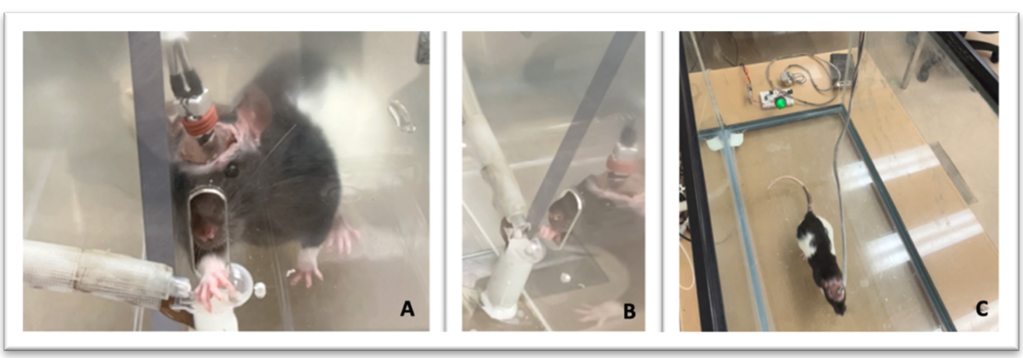


*- Figure 8: **(A & B)** Different setup tested for the LCS configuration. We can also see on the table of the top picture the chessboard used to calibrate the cameras.*

a) Multi-Camera Video Tracking LCS -- V1

- Parameters for Free-behaving Setup:
    - Dimensions of environment: 1m x 1m
    - Number of cameras tested: 2
    - Camera type: iPhone 12-13
    - Camera's Geometry: See picture
    - Frame rate: 60 Hz & 240 Hz
    - Resolution: 1080p HD
    - Calibration: Checkerboard Method
    - Camera Synchronization: Audio channel
    - Neural data & Frame synchronization: Circuit with LED
    - Method of inferring kinematics vector: Pre-trained CNNs Models
    - Method of inferring behaviour type nature: Manual

Above are the main parameters of our Label Collection System. The accurate synchronization of the neural data with the captured frames was achieved by a custom circuit that had the role to output voltage pulses during the recordings (TTL triggers) to a visible LED in the field of view of our video-capture cameras (See Figure 9 C). Those pulses were generated and saved through the Neural Recording System Software called "Synapse" made for neurophysiology experiments (Tucker-Davis Technologies).



*- Figure 9: **(A & B)** Food Reaching setup for the LCS where front paw kinematics could be estimated at 240 fps. **(C)** Free-behaving setup for the LCS where the LED responsible for synchronizing the neural data and the video-capture can be seen turned on.*

4.2 -- Data Processing

A) LCS Pre-Processing

- **Kinematics Extraction**

After the video-capture is made using the LCS system, a structured pipeline was designed to take as input the n-camera recordings plus the calibration sequence, synchronize the camera between them using the audio channel, perform the calibration by extracting the intrinsic and extrinsic parameters matrices. Then, we needed label the joints of the rat on some of the frame in order to fine-tune pre-trained Convolutional Neural Networks (CNNs) on our specific task and finalize the training, which then could output the 3D trajectories of the joint's position on the different frames. To perform this pipeline, many methods were investigated. The two notable that took most of our time were Deeplabcut 3D [10] and DANNCE [11]. Although both methods allow for 3D pose estimation of animals in a free-behaving setup, their architectures are different in many ways as Deeplabcut 3D rely on pose tracking in 2D followed by a triangulation process that generate the 3D pose from multiple cameras. On the other hand, DANNCE learns pose estimation end to end in 3D from all camera views via a method where a single volume serves as input to a 3D convolutional neural network (3D CNN).

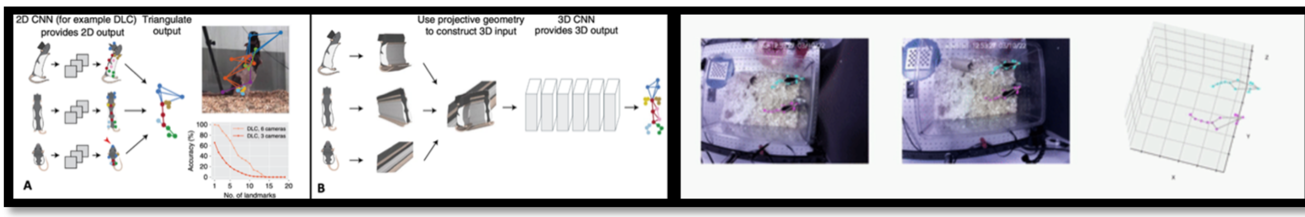


*Figure 10: To the **left**, in **(A)**, we can see the concept of 2D CNN + Triangulation used in more traditional methods like Deeplabcut 3D while in **(B)**, we can see the concept of using a 3D volume as input to a 3D CNN similar to how DANNCE operates. To the **right**, we can a 3D Deeplabcut example of 3D markerless pose estimation on free behaving rodents.*

We tested both methods and got better results with 3D Deeplabcut as DANNCE initially provided us with results containing a lot of noise. And as we were limited in time, we did not have time to troubleshoot the problem and went with Deeplabcut, which has the major advantage of having much more documentation for public usage. Now the last step is to convert those 3D joint's trajectories into the desired decoded variables (ex: acceleration and velocity vectors, relative changes in joint's angles, ...).

- **Behavior Types Extraction**

The behavior extraction was done manually through the analysis of all video sequences. For the classification task, we had two choices to extract either the micro or macro behaviors for the decoding. In other words, this means that should we decode movements done on a very short temporal scale (ex: head turns, paw dynamics, ...) or behaviors done through a longer temporal scale on the order of seconds instead of milliseconds. We decided to go with the ladder in the first place, but when it came to experiment with the former, we found that precise isolated micro-behaviours like the movement of a single leg were on average 0.2 seconds, which produced sparse firing rates vectors across most temporal resolution that made sense. This did not provide us enough information for decoding; therefore, we stayed in the macro-behaviour regime of 2 seconds and implemented a second classifier with 3 behaviours on the second iteration instead of 2 on the first one. Also, we initially had a third behaviours in the V1 Classifier (rodent's grooming), but we did not have enough samples to include it (N < 10), so we discarded it from the design. Furthermore, we avoided all behaviours with vigorous head shakes or cable movements as this may induce us to decode the artifacts and not the motor cortex activity patterns. Now here is an overview of the final behaviours present in the classifiers implemented:

Classifier V1 -- Macro-behavior Decoding

Behavior 1: Rat's standing on its back feet	(N = 44) (Temporal Scale = 2 seconds)
Behavior 2: Locomotion for a minimum of 5 steps	(N = 31) (Temporal Scale = 2 seconds)

Classifier V2 -- Macro-behavior Decoding

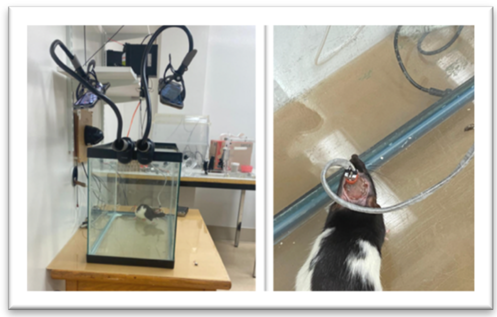
Behavior 1: Rat's standing up on its back feet	(N = 44) (Temporal Scale = 2 seconds)
Behavior 2: Locomotion for a minimum of 5 steps	(N = 31) (Temporal Scale = 2 seconds)
Behavior 3: Food Reaching with front paws	(N = 14) (Temporal Scale = 2 seconds)

B) Neural Data Recording and Pre-Processing

- ### Description of the Neural Recording System:

The Neural Recording System seen on the right figure was used to record, process and store the neural signals. Intracortical signals were amplified (60 dB gain) and bandpass filtered (500 Hz - 5 kHz) and acquired at 25 kHz. The activity on each channel was thresholded using specific parameters on the software (Ex: STD, Polarity). When the threshold was crossed, detected events and their timestamps were saved to disk. When it comes to the microelectrodes used, the rat #191 was implanted a Neuronexus Linear 16x1 (A16x1-2mm-100-177-Z16) (Seen on the right). The rat #194 was implanted a TDT Microwire Array 32 channels. But when it came to the data used in our model, we only used the experimental sessions with rat #194.

- ### Experimentation:



We did 3 experimental sessions on a single day at Dr. Éthier's laboratory, which is in the CERVO research center in Quebec. We first prepared the environment by setting up the mounts and the cameras, followed by the calibration of the ladder with the checkerboard. After that, Dr. Éthier and his team prepared the rat and the recording system. A "fake brain" acting as a synthetic data generator was used to test and verify the quality of the signals. Indeed, this Fake Brain acts like a kind of amplifier and streams generated signals to the system. Then the video-capture was started, the rat was plugged via a wire and put in the arena so the recording could start. After around 20 minutes of free-behaving activity, the recordings were stopped, and their content was saved on disk. From the original 32 channels available for neural recording in the motor cortex, 26 were functional.

	Rat #	Setup	Polarity Threshold	STD Spike Detection	Spike Sorting
Experiment 1	Rat #191	Free Behaving	Negative	x4 STD	Yes
Experiment 2	Rat #191	Free Behaving	Positive	x4 STD	Yes
Experiment 3	Rat #191	Food-Reaching	Positive	x4 STD	Yes

- ### Signal Processing for Neural Decoding:

The software we used performed the filtering and thresholding necessary for spike detection. Several Notch filters were embedded in the software to remove some unwanted frequencies (e.g., 60 Hz). Concerning detection runtime, spikes were detected based on calculations of the deviation of a waveform from its RMS. In the software, we could tune the threshold level in number of standard deviations from the baseline and polarity (either positive or negative). For the first experiment, we set a negative polarity threshold at 4.0 STD level and for the two others, we set a positive polarity threshold at 4.0 STD level. More information on the signal processing performed by the software Synapse is available through their reference manual [12]. Although spike sporting was integrated in the software, we did not make use of it when came time to implement the decoders.

4.3 - Architecture for Neural Decoders

A) Neural Decoding Task

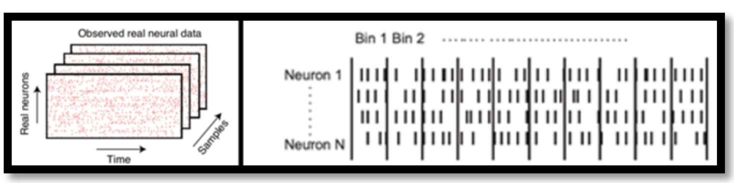
The neural decoding task simply represents a supervised learning mapping where the input vectors or the sequence of input vectors are mapped to the decoded variables, which can be joint's kinematics/angles trajectories, or a probability distribution on the different behaviours. Although we were initially going for the continuous kinematics decoding, we decided to first implement a neural decoder that classifies the different behaviours selected as this was more straightforward to do at first and requires less information content from the recorded channels. Indeed, we experimented with decoders having 26 channels of neural spiking and less than 20 minutes of footage for behaviour extraction each.

Time binning was used to discretize the raw spike trains according to a Temporal Resolution Parameter R. For example, that allowed to transform the neural data of a 2 second behaviour into 20 bins of length R = 100ms. Therefore, if T is the length of the recording, we will have around T/R total data points of neural activity. Now in the case of continuous kinematics decoding, where predictions are made for each output bin, we made sure that we will select an appropriate number of neural data bins that preceded the output as it is logical to do in motor decoding while the inverse is done in the case of sensory decoding. On the hand, only the last probability distribution over the behaviours for all time bins was taken into account with a classification decoder in the case of recurrent neural network inference.

The information content in the neural activity was represented according to the Rate Coding model, where we extracted for each Time Bin the average firing rate of the neuron. Investigating the performances of Temporal Coding or Inter-spike interval (ISI) variables could be something of interest if we manage to have time for it. Hence time binning was used to discretize the raw spike trains according to a Temporal Resolution Parameter R, where each bin was represented by a vector of firing rate for all the neurons/channels.

B) Decoding Pipeline

Before training any model, we first filtered and removed all noisy/non-functioning channels, which in our case represented 6 channels of the 32 available. Now for behaviour classification decoding, after the frames were synchronized with the neural data and the different timesteps (Start-End) of the behaviours were extracted, we build a pipeline to automatically extract a dataset with the format of a 3D tensor with dimensions B x T x N. The B represents the number of sample behaviours in the dataset (Number of classes x respective counts), the T represents the number of Time Bins selected via the Temporal Resolution parameter, and finally N stands for the number of neurons/channels kept. This allowed us to take raw spike trains via the detected events of the recording system and transform them rapidly into neural decoding datasets composed of a sequence of firing rates vectors that correspond to a specific behaviour sample. Also, the target vector representing the probability distributions over all possible behaviours was generated for each sample via a one-hot encoding scheme.



*- Figure 11: To the **left**, we can see raw firing train samples. To the **right**, we can see the process of discretization of the signal into firing rates vectors via time binning the raw neural data.*

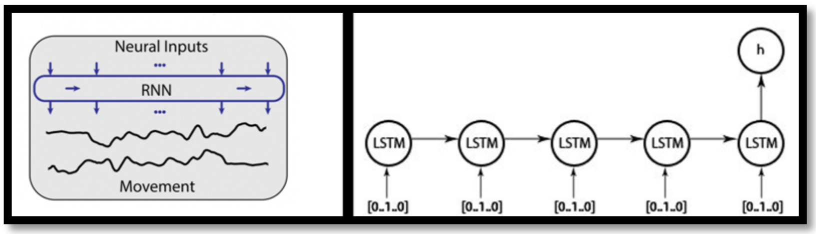
	Nb of Behaviours	Total Tensor size	Temporal Resolution R / Bin	Total Bins	Nb Channels	Dataset Size
Classifier - Decoder V1	2	26 x 20 x 75	100 ms	20	26	75
Classifier - Decoder V2	3	26 x 20 x 75	100 ms	20	26	89

- Table: Overview of the parameters of the 2 different decoders implemented on two different datasets generated from experiment #1.

5) Models, Training & Results

- Models

The first architecture tested for our decoder V1 was the Long-Short Term Memory (LSTM) model, a powerful recurrent network that is well adapted to model sequences of vectors, whether those are word embeddings in NLP, speech, or neural firing rates vectors like in our case. In fact, most of the best results in neural decoding are often achieved with this kind of architecture as they scale well with more data and can represent complex non-linear relations across time without problems. This is the case for both classification and regression based neural decoders.



*- Figure 12: To the **left**, we can see a continuous RNN decoder regressing movement trajectories. To the **right**, we can see an LSTM used to predict a single probability distribution at the end, which is mostly similar to our case where the input vectors at different time steps are the firing rate vectors and the output is the probability distribution over the different behaviour classes.*

We tested multiple configurations of this architecture by tuning different hyperparameters and compared the performances between versions (Additional details are presented below). Also, we compared the LSTM with other types of model in order to assess the accuracy of different approaches, but the LSTM seemed always superior.

Although using 26 neural data channels is not something large for modern neural decoder, applying a dimensionality reduction method could have strong benefit when having large channel counts as this may allow the compress of information content and keep only what is relevant to the task. Indeed, this concept related to the topic of Neural Manifold decoding, where a reduction in the neural state space via linear methods (Ex: PCA) or non-linear ones like Autoencoders can produce a low-dimensional neural manifold that is shaped by the covariance between neurons. And according to recent research, it is proposed that spikes may be a manifestation of the latent dynamics of a smaller number of latent variables which resides in this low-dimensional space. Therefore, decoding motor signals directly from the latent variables manifold may be more optimal depending on the number of recorded channel available and the complexity of the task decoded. In our case, we decided that we would experiment with those ideas if we had time to do.

- Training & Results

For the V1 decoder, a sequence of 20 firing rate vectors corresponding to 20 bins of 100ms activity (Total: 2000 ms) was processed in the LSTM model, where after transitioning through all bins, the final hidden state generated was used to predict the output probability distribution via a softmax function. Hence, this allowed the integration of neural sequences over an extended period of time in order to produce the final probability of detecting a specific behaviour. The loss function was categorical cross entropy due to the multi-class classification task and the training was performed via the parameter specific in the table below.

Before any training, we normalized our data between [0-1] as LSTMs are known to be sensitive to the scale of the input data. And more importantly, due to the relatively small size of our dataset and thus the high number of epochs needed to allow convergence of the optimization process, we used regularization via the dropout parameter that allowed us to tune the proportion of units that get dropped out to avoid a strong overfitting.

Models	Accuracy	Test/Training Ratio	Regularization	No. of epoch	Hidden units	Activation Function
LSTM - V1	≈ 82%	0.2	0.2	500	400	Softplus (Smooth ReLu)
LSTM - V2	≈ 70%	0.2	0.2	500	500	Softplus (Smooth ReLu)

Table: Results and hyperparameters for the 2 main neural decoders using the LSTM architecture.

6) Analysis & Conclusion

Looking at the result's table above, we can see that we successfully decoded information content from the motor cortex activity of a rat in order to classify with a reasonably good accuracy a small part of his behaviour. Indeed, the LSTM -- V1 trained to predict the accurate probability distribution over 2 behaviours from a temporal sequence of 20 firing rates vectors contained in 2 seconds of neural activity has achieved an overall accuracy of 82% on the test set (on average). And for the LSTM -- V2, which was trained to predict in this case the probability distribution over 3 behaviours, we achieved an accuracy of 70% on the test set (on average). But due to the fact that we trained the last model with large differences in the number of samples for each class (44, 31, 14), we are cautious in interpreting the results obtained. For a next iteration, collecting more data to balance the class counts should be better or at least making sure the count is more balanced.

Due to the relatively small size of our dataset (N = 75 for V1), we made sure to randomly shuffle the making of the training and test set in the pipeline. Furthermore, we increased the size of the test set from 10% to 20% and performed the training/inference process multiple times in order to make sure the accuracy on the test set is valid. Moreover, implementing a K-fold validation to enhance the validity of our results is something we were looking to do if not constrained in time. Of course, we only used data from a single experiment and a small dataset was used to train the neural decoder. But we are well aware that this decoder is only symbolic as this project was designed to gain experience in neural decoders and the related signal processing & machine learning. We could have pushed the experimentation further as we had the possibility to return for more recording sessions at CERVO and produce more distinct behaviours, but our purpose was to successfully decode from the motor cortex some relevant information and learn a lot during the process. Also, taking more time and resources from Dr. Ethier laboratory at CERVO was not something necessary according to us. Now although we did not have time to implement many concepts we had initially, like kinematics decoding or using the Transformer architecture in the decoding process, we nonetheless think that we produced something interesting for this project.

- Review on the Objective

In retrospective, we can affirm that we investigated the possibility of decoding from the motor cortex of a free-behaving rodent some corresponding behaviours via first, the design and usage of a label collection system (LCS) for collecting the target behaviours of rats while recording their motor cortex activity using micro-electrode arrays implanted by Dr. Ethier and his laboratory. Then, we build a pipeline for processing and generating datasets, followed by a pipeline for training neural decoders using Sequence-to-Sequence Deep Learning architecture like the LSTM. Like explained earlier, we nonetheless think we deviated a bit from our original goal as we switched to a classification task as decoding precise trajectories across time requires much more information content in the signals. Although we were interested in investigating the accuracy and impact of using different temporal scale for the decoded behaviours (micro vs macro), as soon as we considered behaviours under the 1 second scale with a temporal reasonable temporal resolution, the firing vector matrices became too sparse, which kept us experimenting in the macro-behaviour regime. Nonetheless, we think that the question of whether decoders should operate in the micro or in the macro behaviour regime when decoding complex sequences of motor information is still interesting.

- Looking into the future

Now looking at the future of neural decoding and the various research avenues proposed in the introduction, getting enough diverse data for training accurately large approximative models will require the use of automated pipeline that will allow continual learning for the decoders by continuous collection of large amounts of time made of various mapping between neural patterns and the target behaviour. Label Collection System (LCS) implemented with free behaving subjects may be a solution to the traditional training of animals to perform the desired tasks for weeks. Although we implemented a video based LCS, we knew that it is not the scalable one as the data collection need to be much longer and in a wide range of environment without constraining physically the subject. Hence, LCS using alternative designs like the usage of precise sensors-embedded suite or many other ideas we had were the best alternative for long term development of system collecting labels for training accurate sensorimotor neural decoders and encoders. Combining those kinds of ideas with methods like synthetic data augmentation may allow one day the multi-contextual generalization of the process through which it is possible to encode and decode information from the neocortex.

- Acknowledgment

We thank Dr. Christian Ethier (Laval University) and his laboratory personnel (Maxime) at CERVO for the help provided to realize this project. We also thank Dr. Gosselin, Dr. Guillaume, and Mme. Michelle for approving this unorthodox project in the context of a graduate course in Electrical Engineering at Laval University.

REFERENCES

[1] The Argo: A 65,536 channel recording system for high density neural recording in vivo

[2] An integrated brain-machine interface platform with thousands of channels

[3] Massively parallel microwire arrays integrated with CMOS chips for neural recording

[4] A versatile robotic platform for the design of natural, three-dimensional reaching and grasping tasks in monkeys. B Barra et al 2020 J. Neural Eng. 17 016004

[5] Jesse D. Marshall, Diego E. Aldarondo, Timothy W. Dunn, William L. Wang, Gordon J. Berman, Bence P. Ölveczky. Continuous Whole-Body 3D Kinematic Recordings across the Rodent Behavioral Repertoire, Neuron, Volume 109, Issue 3, 2021, https://doi.org/10.1016/j.neuron.2020.11.016.

[6] Foster JD, Nuyujukian P, Freifeld O, Ryu SI, Black MJ, Shenoy KV. A framework for relating neural activity to freely moving behavior. Annu Int Conf IEEE Eng Med Biol Soc. 2012;2012:2736-2739. doi:10.1109/EMBC.2012.6346530

[7] Barroso FO, Yoder B, Tentler D, et al. Decoding neural activity to predict rat locomotion using intracortical and epidural arrays. J Neural Eng. 2019;16(3):036005. doi:10.1088/1741-2552/ab0698

[8] Glaser JI, Benjamin AS, Chowdhury RH, Perich MG, Miller LE, Kording KP. Machine Learning for Neural Decoding. eNeuro. 2020 Aug 31;7(4):ENEURO.0506-19.2020. doi: 10.1523/ENEURO.0506-19.2020. PMID: 32737181; PMCID: PMC7470933.

[9] Neuralink Show and Tell, Fall 2022. https://youtu.be/YreDYmXTYi4

[10] Nath, T., Mathis, A., Chen, A.C. et al. Using DeepLabCut for 3D markerless pose estimation across species and behaviors. Nat Protoc 14, 2152–2176 (2019). https://doi.org/10.1038/s41596-019-0176-0

[11] Dunn TW, Marshall JD, Severson KS, et al. Geometric deep learning enables 3D kinematic profiling across species and environments. Nat Methods. 2021;18(5):564-573. doi:10.1038/s41592-021-01106-6

[12] Synapse Manual, TDT. https://www.tdt.com/files/manuals/SynapseManual.pdf