Towards a Neural Co-Processor Which Restores Movement After Stroke: Modeling a Proof-of-Concept

Matthew J Bryan[1], **Linxing Preston Jiang**[1], **Rajesh P N Rao**[1]

[1] Neural Systems Laboratory, Department of Computer Science and Engineering, University of Washington, Box 352350, Seattle, WA 98105, USA

E-mail: matthew.bryan@dell.com

September 2021

Abstract. *Objective* Brain co-processors[1] are devices which use artificial intelligence (AI) for closed-loop neurostimulation, to shape neural activity and to bridge injured neural circuits for targeted repair and rehabilitation. The co-processor framework offers a flexible approach to learning closed-loop stimulation policies that optimize for (a) specific regimes of neural activity, or (b) external task performance. For example, it may seek to learn to stimulate the motor cortex of a stroke patient, conditioning the stimulation on upstream visual information, aiding the patient's attempt to grasp an object. Through the use of artificial neural networks (ANNs) and deep learning, the co-processor co-adapts with the neural circuit, allowing it to seek optimal stimulation policies, and adapt them as the neural circuit changes. The results presented here demonstrate a neural co-processor for the first time, through the use of a simulation. We explore some of the core algorithms that may allow co-processors to successfully learn and to adapt to non-stationarity in both the brain and sensors. *Approach* We provide the first proof-of-concept of a neural co-processor that leverages deep learning, through the use of a simulated neural circuit That circuit performs reach-to-grasp task, based on visual input, and is designed to closely resemble a similar circuit in a primate brain [2]. We simulate a variety of lesions by altering the model, and then demonstrate a co-processor's ability to restore lost function through "stimulation" of that model. We further test the ability of a co-processor to adapt its stimulation as the simulated brain undergoes changes. *Main results* Our simulated co-processor successfully co-adapts with the neural circuit to accomplish the external reaching task. The co-processor framework demonstrated here adapts to a variety of lesion types, and to ongoing changes in the simulated brain. *Significance* The proof-of-concept here outlines a co-processor model, as well as our approach to training it, leading to insights on how such a model may be developed for *in vivo* use. We believe this co-processor design will allow for learning complex stimulation policies that help restore function to a stroke victim.

Keywords: brain-computer interface, neural co-processor, ai, machine learning, stimulation

1. Introduction

Aided in part by application of advanced AI techniques, brain-computer interfaces (BCIs) have made advancements over the last several decads, allowing for decoded brain signals to be used for control of a wide variety of virtual and physical prostheses [3, 4, 5, 6]. Separately: advances in stimulation techniques and modeling have allowed us to probe neural circuit dynamics (e.g. [7]) and learn to better drive neural circuits towards target dynamics, by encoding and delivering information through stimulation [8, 9, 10, 11, 12, 13, 14, 15]. Recently, there has been increasing interest in building on these advances to combine decoding and encoding in a single system, for closed-loop stimulation of a neural circuit. Bi-directional BCIs (BBCIs) allow stimulation to be conditioned by decoded brain activity as well as external sensor data (e.g. camera), which can allow for the application of real-time, fine-grained control of neural circuits and prosthetic devices, e.g. Nicolelis et al. [16]. These may lead, for example, to neuro-prostheses that are capable of restoring movement which was lost due to traumatic brain injury (TBI), to a degree not previously possible.

Motivated by that progress, we demonstrate here a flexible framework for combining encoding and decoding, which we term "neural co-processors" [1]. Neural co-processors leverage AI and deep learning to identify optimal, closed-loop stimulation patterns. The approach is flexible enough to optimize not only for particular neural activities, but also for tasks external to the subject. For example, they may be able to aid a stroke victim by finding a stimulation pattern of the motor cortex which helps restore lost limb function. Likewise, the framework generalizes enough to condition stimulation on both brain activity, and external sensors, e.g. cameras or LIDAR, in order to incorporate feedback for realtime control.

Additionally, the co-processor framework allows a neuro-prosthesis to actively adapt to a neural circuit as it changes with time. This framework is capable of co-adapting with the circuit, i.e. brain, by updating its stimulation regime, while at the same time the brain is updating its response to the stimulation, and changing due to natural plasticity, aging, etc. This allows the co-processor to continually optimize for the intended cost function, despite the signficant non-stationarity of the target circuit.

Here we provide a proof-of-concept in simulation for a co-processor that restores movement to a limb, after a subject has suffered a stroke affecting its ability to use that limb. It combines:

- A stimulation model, which models the relationship between decoded brain activity, stimulation, and task performance.
- An AI agent which determines the stimulation to apply in a closed-loop fashion, in real time.

Significant advances have been made in modeling the effects of electrical stimulation of the brain, some of which can be leveraged for our co-processor design, as we outline below. Researchers have explored how information can be biomimetically or artificially

encoded and delivered via stimulation to neuronal networks in the brain and other regions of the nervous system for auditory [8], visual [9], proprioceptive [10], and tactile [11, 12, 13, 14, 15] perception. Advancements have also been made in modeling the effects of stimulation over large scale, multi-region networks, and across time [17]. Some have additionally designed models which can adapt to ongoing changes in the brain, including changes due to the stimulation itself [18]. In our proof-of-concept outlined below, we will use a stimulation model, not unlike those cited here, which seeks to account for both network dynamics and non-stationarity. In addition to training the model to have a strong ability to predict the effect of stimulation, we additionally train it to be useful for then learning an optimal stimulation policy, which is a property somewhat distinct from predictive power alone.

Advances have also been made in both open- and closed-loop stimulation for treating a variety of disorders. Open loop stimulation has been effective in treating Parkinson's Disease [19], as well as various psychiatric disorders [20, 21, 22]. More directly related to this paper, we see in Khanna et al. [23], the use of open loop stimulation in restoring dexterity after a lesion occurs affecting a primate's motor cortex. The authors demonstrate that the use of low-frequency alternating current, applied epidurally and set to certain phases, can improve grasp performance.

While open loop stimulation techniques have yielded clinically useful results, their results in many domains have been mixed, such as use in visual prostheses [24], and use in invoking somatosensory feedback [15]. Likely this is due to the stimulation not being conditioned on the ongoing dynamics of the circuit being stimulated. Moment-to-moment and throughout the day, the circuit will respond differently to the same stimulus, as a result of differing inputs and ongoing activity. Stimulation therefore needs to be proactively adapted in response. This need is even greater over longer time scales as the effects of plasticity and ageing change the connectivity of the brain.

Closed-loop stimulation conditions stimulation on observations of brain activity, possibly allowing it to shape the neural activity more precisely, and to adapt to changes in the circuit over time. This opens the door to real-time, targeted control of the neural circuit. It has been used to aid in learning new memories after some impairment [25, 26], to replay visually-invoked activations [18], and for optogenetic control of a thalamocortical circuit [27], among others.

Something that remains unclear is how to leverage closed-loop control for real-time co-adaption with the brain to accomplish an external task. "Co-adaption" here refers to the ability of a neuro-prosthesis to adapt its stimulation regime to the ongoing changes in the circuit it is stimulating, and to adapt with that circuit to accomplish the external task, such as grasping. The neural co-processor we present here provides one potential model for accomplishing that. Through the use of deep learning, the co-processor model we present co-adapts an AI agent, which governs the stimulation, with both a stimulation model, and the neural circuit being stimulated.

For a neurologically complex task such as grasping, we cannot identify *a priori* a real time controller of the neural circuit. That is due in large part to the variability of

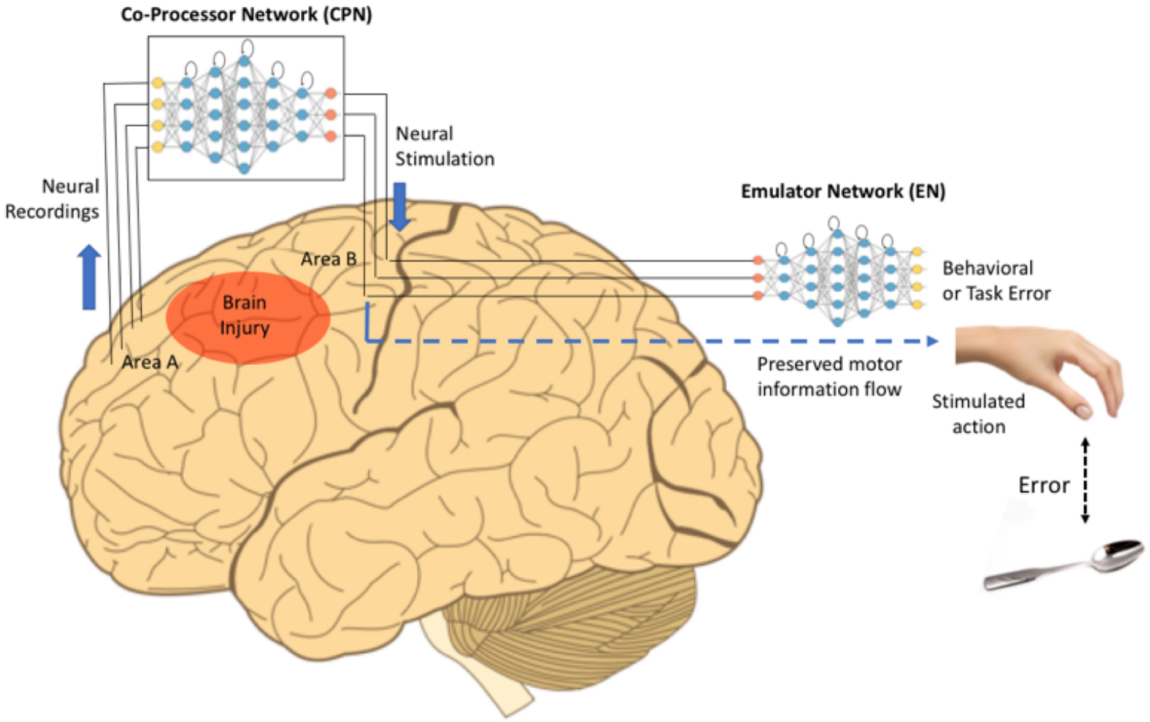

Figure 1. Using a co-processor to drive external task performance after a traumatic brain injury

circuits from subject-to-subject, as well as variations in the placement of sensors and stimulators in the brain. The only plausible path to such a real time controller is to parameterize it in a subject- and time-specific way. Our model seeks to accomplish that using deep learning, together with a data-efficient approach to training.

Before attempting *in vivo* experiments using such a model, we first demonstrate here a number of crucial design elements of it, through the use of a simulated grasping circuit, presented previously by Michaels et al. [2]. We explore:

- The properties of the artificial neural networks (ANNs) that are needed to successfully adapt to the long-running dynamics of a stimulated neural circuit, as well as to adapt to that circuit's ongoing changes.
- Data-efficient methods for training these models, to better ensure we can train them with a biologically-realistic amount of data.
- How we must train our stimulation model to make it effective in later training our stimulation agent.

2. Method

2.1. Architecture Overview

First, we present the architecture of our co-processor design. This design aims to solve two fundamental challenges in using neural stimulation to improve external

task performance. First, neural networks exhibit long-running and highly nonlinear dependencies, necessitating the need for a stimulation agent to account for far-distant effects of the stimulation it applies. Second, with neurologically complex tasks, such as grasping, the mapping between neural activity as-measured and the external task cannot be determined *a priori*. As a result, the co-processor must somehow learn what stimulation is appropriate for aiding the user in the external task they are attempting to perform.

Our co-processor attempts to solve these with a pair of artificial neural networks:

- A stimulation and neural dynamics model, known as an "Emulator Network" (EN). It models the relationship between the stimulation, neural dynamics, and external task. It's purpose is for training the second network.
- A stimulation agent, known as the "co-processor network" (CPN), which maps neural activity, and possibly data from external sensors, to stimulation parameters.

We co-train the EN and CPN, with the goal of training a CPN whose output stimulation parameters improve task performance. By continually training both, they adapt to the brain as it changes, effectively allowing for brain-stimulator co-adaptation. The EN is a tool for training the CPN, giving us a way to back-propagate task error to the CPN. It outputs task-relevant metrics - a prediction of muscle velocities in our case - given measurements of neural activity, and the stimulation parameters output from the CPN. If the EN is trained in a particular way, and to a sufficient level of precision, it can be used as a function approximator relating stimulation and a task, and do so in a way that allows us to train the CPN with it. When training the CPN, we in-effect treat the EN's output as the true task performance, or a related metric, and then backpropagate the loss defined in terms of that metric in order to train the CPN. See Fig. 1. We will illustrate the details of the training algorithm below.

In our present demonstration, the EN is constituted as a single layer, fully connected, long short-term memory (LSTM) recurrent neural network (RNN), with hyperbolic tangent ($tanh$) activations, and a linear readout. The general notion of a co-processor does not require this precise architecture, but for our example, we found that the LSTM approach allows the network to continuously adapt to long-running dependencies in the simulated neural dynamics, far better than a vanilla RNN. The CPN is constituted as an almost identical network, though with a different dimensionality, as we explain below. There is no strict requirement for the EN and CPN to be so similar, but we found this simple architecture to work well in our example.

Note that the EN effectively constitutes a stimulation model. Its design is somewhat motivated by the common linear time-invariant state space model of stimulation, as in e.g. Yang et al. [17], which is a helpfully simple (linear) approach. That approach seeks to model neural network dynamics in terms of a stationary linear model, which, due in part to its simplicity and the number of techniques developed for it, lends itself to useful interpretations and analyses of those neural dynamics.

There are some key differences between that approach and the EN architecture we

present here. Our EN's purpose is simply to train the CPN, so we pick an architecture we find to be best suited to that job. First, our EN architecture is not linear, through the use of a *tanh* activation function. That allows us to escape some of the limits in representational power of linear models. Second, LSTM cells are designed to better capture long-running network dynamics, and in our case we found that they were crucial for that reason. Compare that to the simpler cells commonly used in vanilla RNNs, which yield functionally the same approach as the linear state space model, less the use of a nonlinear activation function.

// TODO? Try having no nonlinearity?

2.2. Simulation Overview

* Proof-of-concept by way of simulation. ** Simulation allows for rapid, cheap exploration of learning algorithms. ** Potentially helpful to look for a simulated neural circuit which has internal dynamics, i.e. information propagation, which is demonstrably similar to natural circuits. ** [28] Likeness of RNNs to natural circuits re: dynamics ** Cite Michaels, and those upstream from him (Susillo?) re: RNNs. Networks have internal dynamics that result from both inputs and internal connectivity. Perturbations at t=0 affect outputs at t=N. ** [29] Simulation of spiking neural network, learning stimulation regime. Use of a biomimetic simulation to develop and evaluate a neural controller. Use of lesion model.

The stimulation model in our case adapts itself to the task of training our closed-loop AI stimulation agent. For a neurologically complex task such as grasping, we don't know *a priori* what stimulation regimes to to explore while searching for an optimal solution. Our stimulation model plays the role of generating learning signals, via error backpropagation (¡TODO? Citation?¿), for our stimulation agent. It continuously updates itself based on observations of brain activity, stimulation, and external task queues, such as hand position.

Example: Use of actual stimulation of an actual brain, together with offline deep learning, to search for an optimal visual input: [7]

* Overview of test bed ** Michaels model [2] [30]. ** Lesion designs *** Lesion examples, i.e. hand velocity more affected than shoulder velocity. *** Disconnect modules vs lesion M1, and the effects ** Stimulation design *** Spatio-temporal smoothing ** Observation model

* Overview of the four experiments: ** Vanilla ** Co-adaptive with connectivity loss ** Co-adaptive with partial M1 loss ** Co-adaptive with data reuse

2.3. Training the Model

Training ** Alternate training EN and CPN *** Train an EN **** Training the EN until predictive power reaches a threshold, based on current task performance **** Training regime based on most recent CPN, similar CPNS (noise added to params), and white noise stimulation **** Single batch of data from applying current CPN set

and white noise to the brain, EN fit to "offline" it over maining epochs **** Predictive power alone is not sufficient for use in training the CPN: we need to form the batch as above, otherwise training is unstable. *** Train the CPN **** Backprop through the EN, effectively using the EN's prediction as ground truth

3. Results

* Task performance improves drastically * Fine tuning takes a long time * Object classes separate * Can co-adapt with the brain, as it changes

* Split by lesion design

* Show example where AIP/F5 lesioned and how that affects performance

* Any point in showing sensitivity to observation or stim dim?

* Try regularization variation?

* White noise and noised params ** Learning instability ** Predictive power isn't enough ** Illustrate this

* Training efficiency analysis

4. Discussion

* Training efficiency * Spectrum from simple low dimensional stimulation vectors today to higher dimensional future * Toward in vivo application

5. Conclusion

asdf

6. Acknowledgements

Ganguly, Priya, Anca, Justin, Luciano

7. Ethical Statement

asdf

8. References

[1] RPN R 2019 *Current Opinion in Neurobiology* **55** 142–151

[2] Michaels J, Schaffelhofer S, Agudelo-Toro A and Scherberger H 2020 *Proceedings of the National Academy of Sciences* **117** 32124–32135 ISSN 0027-8424 (*Preprint* `https://www.pnas.org/content/117/50/32124.full.pdf`) URL `https://www.pnas.org/content/117/50/32124`

[3] Rao R 2013 *Brain-Computer Interfacing: An Introduction* (Cambridge University Press) ISBN 9780521769419

[4] Wolpaw J and EW W 2012 *Brain-Computer Interfaces: Principles and Practice* (Oxford University Press)

[5] Moritz C, Ruther P, Goering S, Stett A, Ball T, Burgard W, Chudler E and Rao R 2016 *IEEE transactions on bio-medical engineering.* **63**(7) 1354–1367

[6] Lebedev M and Nicolelis M 2017 *Physiological reviews* **97**(2) 767–837

[7] Walker E e a 2019 *Nature Neuroscience* **22**(12) 2060–2065 URL https://doi.org/10.1038/s41593-019-0517-x

[8] Niparko J 2009 *Lippincott Williams and Wilkins* (Oxford University Press)

[9] Weiland J and Humayun M 2014 *IEEE transactions on bio-medical engineering* **61**(5) 1412–1424

[10] Tomlinson T and Miller L 2016 *Advances in experimental medicine and biology* **957** 367–388

[11] Tabot G A, Dammann J F, Berg J A, Tenore F V, Boback J L, Vogelstein R J and Bensmaia S J 2013 *Proceedings of the National Academy of Sciences* **110** 18279–18284 ISSN 0027-8424 (*Preprint* https://www.pnas.org/content/110/45/18279.full.pdf) URL https://www.pnas.org/content/110/45/18279

[12] Tyler D 2015 *Current opinion in neurology* **28**(6) 574–581

[13] Dadarlat M, O'Doherty J and Sabes P 2015 *Nature neuroscience* **18**(1) 138–144

[14] Flesher S N, Collinger J L, Foldes S T, Weiss J M, Downey J E, Tyler-Kabara E C, Bensmaia S J, Schwartz A B, Boninger M L and Gaunt R A 2016 *Science Translational Medicine* **8** 361ra141–361ra141 (*Preprint* https://www.science.org/doi/pdf/10.1126/scitranslmed.aaf8083) URL https://www.science.org/doi/abs/10.1126/scitranslmed.aaf8083

[15] Cronin J, Wu J, Collins K, Sarma D, Rao R, Ojemann J and Olson J 2016 *IEEE transactions on haptics* **9**(4) 515–522

[16] O'Doherty J, Lebedev M, Ifft P, Zhuang K, Shokur S, Bleuler H and Nicolelis M 2011 *Nature* **479**(7372) 228–231 URL https://doi.org/10.1038/nature10489

[17] Yang Y, Qiao S, Sani O, Sedillo J, Ferrentino B, Pesaran B and Shanechi M 2021 *Nature Biomedical Engineering* **5**(4) 324–345 URL https://doi.org/10.1038/s41551-020-00666-w

[18] Tafazoli S, MacDowell C, Che Z, Letai K, Steinhardt C and Buschman T 2020 *Journal of Neural Engineering* **17** 056007 URL https://doi.org/10.1088/1741-2552/abb860

[19] Benabid A 2003 *Current opinion in neurobiology* **13**(6) 696–706

[20] Holtzheimer P and Mayberg H 2011 *Annual review of neuroscience* **34** 289–307

[21] Kisely S, Hall K, Siskind D, Frater J, Olson S and Crompton D 2014 *Psychological medicine* **44**(16) 3533–3542

[22] Fraint A and Pal G 2015 *Frontiers in Neurology* **6** 170 ISSN 1664-2295 URL https://www.frontiersin.org/article/10.3389/fneur.2015.00170

[23] Khanna P, Totten D, Novik L, Roberts J, Morecraft R and Ganguly K 2021 *Cell* **184**(4) 912–930 URL https://pubmed.ncbi.nlm.nih.gov/33571430/

[24] Bosking W, Beauchamp M and Yoshor D 2017 *Annual review of vision science* **3** 141–166

[25] Berger T, Song D, Chan R, Marmarelis V, LaCoss J, Wills J, Hampson R, Deadwyler S and Granacki J 2012 *IEEE transactions on neural systems and rehabilitation engineering : a publication of the IEEE Engineering in Medicine and Biology Society* **20**(2) 198–211

[26] Kahana M J e a 2021 *medRxiv* (*Preprint* https://www.medrxiv.org/content/early/2021/05/22/2021.05.18.212569 URL https://www.medrxiv.org/content/early/2021/05/22/2021.05.18.21256980

[27] Bolus M, Willats A, Rozell C and Stanley G 2021 *Journal of neural engineering* **18**(3)

[28] Kao J 2019 *Journal of Neurophysiology* **122** 2504–2521 URL https://doi.org/10.1152/jn.00467.2018

[29] Dura-Bernal S, Li K, Neymotin S, Francis J, Principe J and Lytton W 2016 *Frontiers in Neuroscience* **10** 28 ISSN 1662-453X URL https://www.frontiersin.org/article/10.3389/fnins.2016.00028

[30] Sussillo D, Churchland M, Kaufman M and Shenoy K 2015 *Nature Neuroscience* **18**(7) 1025–1033