Supplementary material

Lightweight reinforcement learning decoders for autonomous, scalable, neuromorphic intra-cortical brain machine interfaces

Aayushman Ghosh[1,2,†], Shoeb Shaikh[3,†], Biyan Zhou[4], Pao-Sheng Vincent Sun[4], Camilo Libedinsky[5,6], Rosa Q. So[7,8] and Arindam Basu[4,*]

[1] Department of Electronics and Telecommunication Engineering, Indian Institute of Engineering Science and Technology, Shibpur, India

[2] Department of Electrical and Computer Engineering, University of Illinois Urbana-Champaign, Urbana, Illinois 61801, USA

[3] School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798, Singapore

[4] Department of Electrical Engineering, City University of Hong Kong, 83 Tat Chee Avenue, Kowloon, Hong Kong SAR, Hong Kong

[5] Institute of Molecular and Cell Biology (IMCB), Agency for Science, Technology and Research (A*STAR), Singapore 138673, Singapore

[6] Department of Psychology, National University of Singapore, Singapore 117570, Singapore

[7] Institute for Infocomm Research (I^{2}R), Agency for Science, Technology and Research (A*STAR), Singapore 138632, Singapore

[8] Department of Biomedical Engineering, National University of Singapore, Singapore 117583, Singapore

†These authors contributed equally to this work.

* Correspondence author; E-mail: arinbasu@cityu.edu.hk.

1. Neural dataset simulation methodology

1.1. Izhikevich model

Syn_Dir_4 corresponds to the synthetic neural spike data generated for four-options cursor control reported in experiment 2. We took experiment 2–day 1's target matrix and obtained a neural matrix, $\mathbf{X}$, ($\mathbf{X} \in \mathbb{R}^{T \times \mathscr{D}}$; T-number of time-steps and $\mathscr{D}$–number of input neurons) following the Izhikevich method given by,

$$v' = 0.04v^2 + 5v + 140 - u + I \tag{1a}$$

$$u' = a \cdot (av - u) \tag{1b}$$

with resetting of auxiliary spike,

$$\text{if } v \geq +30\text{mV then } \begin{cases} v \leftarrow c \\ u \leftarrow u + d \end{cases} \tag{2}$$

where, v is the membrane potential and u is the membrane recovery variable, which accounts for the activation of K^+ ionic currents and inactivation of Na^+ ionic currents, and it provides negative feedback to v. After the spike reaches its apex ($+30\ mV$), the membrane voltage and the recovery variable are reset according to the (37). Synaptic currents or injected dc-currents are delivered via the variable I. The parameter a describes the time-scale of the recovery variable u, and the typical value is $a = 0.02$. The The parameter b describes the sensitivity of the recovery variable u to the subthreshold fluctuations of the membrane potential v. The typical value is $b = 0.2$. Here, c refers to after-spike reset value of the membrane potential v, and d refers to after-spike reset of the recovery variable u. The typical value of (c,d) ranges between $(-65,8)+(15,6)\cdot e^2$ where e is a random variable uniformly distributed, $e \in [0,1]$. I is the synaptic current which is calculated from target variable as 1 for spike and 0 for all other times.

1.2. Neural spike datasets

Syn_Dir_4 was generated for four target states corresponding to the four actions—left, right, forward and stop. We considered the number of input neurons to be $\mathscr{D} = 60$, and split them into five ensembles comprising of 12 neurons each. Following the methodology reported in [1], we tuned four ensembles to each of the four output actions, and the fifth ensemble was left uncorrelated with the output action space. The fifth ensemble served to simulate noise and the synaptic current I randomly chose values from a standard Gaussian distribution for these neurons.

I takes on the value of 1 for a neuron tuned to a specific output action (target) or 0 otherwise at every time-step, $t = i$, for the tuned ensemble of neurons. The target value corresponding to every time-step is taken from day 1 of experiment 2. One must note that real world neural data suffers from issues such as electrode deterioration, electrode micro-motion, changes in electrode impedance among others. To account for these effects, authors in [1] propose changing value of a tuned neuron's 1 from $1/0$ to a value chosen at random from standard Gaussian distribution at every time-step, $t = i$. The proportion of such noisy neurons were added in steps of 10% from 0 to 40% to the tuned neuron ensembles in order to create five copies of neural spike data with varying degrees of noise [1]. Added noise introduces variability/non-stationarity in neural data.

Similarly, we created *Syn_Dir_8* for eight output actions corresponding to movement towards the eight center-out targets. In this case, we used target matrix corresponding to day 1 of experiment 4 to arrive at the neural data matrix. We used $\mathscr{D} = 63$ input neurons in this case and split them into nine equal sized ensembles – eight ensembles tuned to each of the eight directions and the remaining one for noise. Furthermore, we introduced additional noise in 0 to 40% of tuned neurons in steps of 10%, by changing the value of synaptic current from $1/0$ to a random value chosen from standard Gaussian distribution. This yields us five versions of neural spike data with varying degrees of noise (variability/non-stationarity).

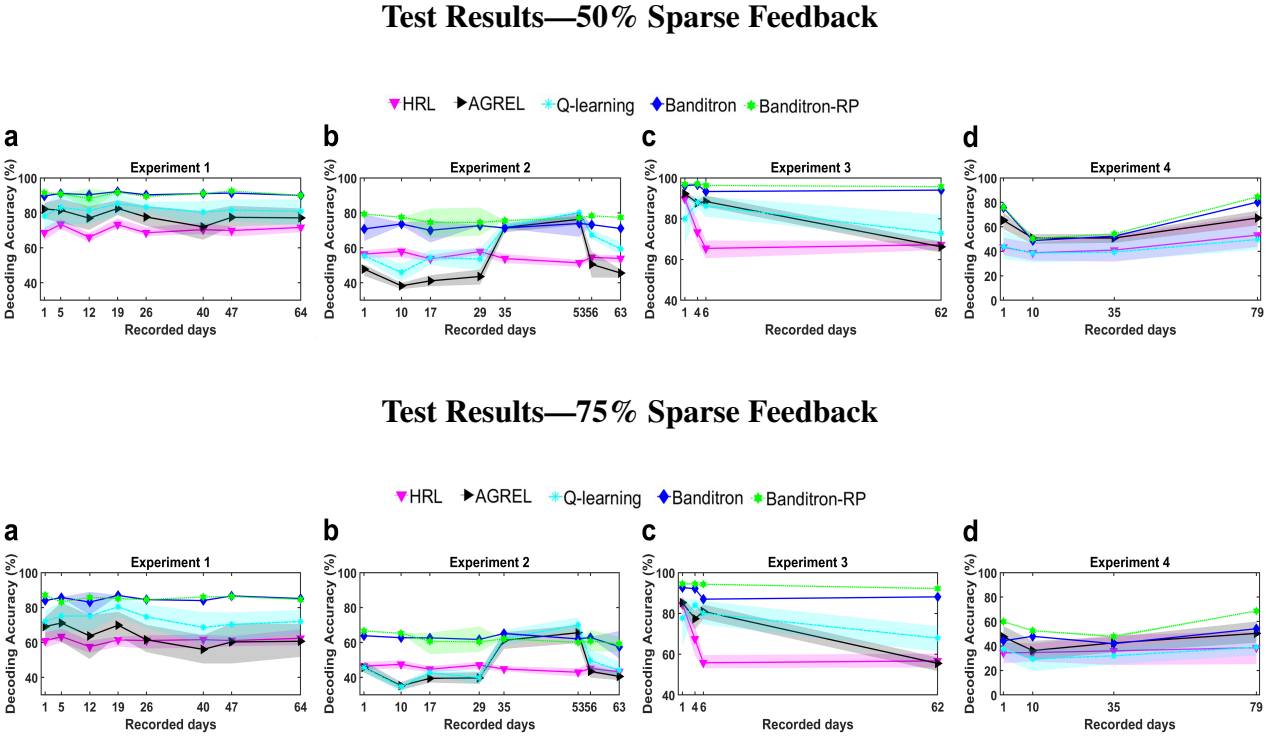

Figure S1. Decoding accuracy across four experiments has been reported for RL algorithms—AGREL, HRL, Banditron, Banditron-RP and Q-learning withholding feedback signal across time-steps thereby introducing sparsity in feedback. **(a)**, **(b)**, **(c)**, **(d)** depict results for 50% sparsity in feedback—First half (top); and **(a)**, **(b)**, **(c)**, **(d)** depict results for 75% sparsity in feedback respectively—Second half (bottom). Shaded regions represent standard deviation of results across 20 iterations of random instantiations of probabilistic algorithms. In this scenario, Banditron and Banditron-RP significantly outperform the state of the art RL algorithms.

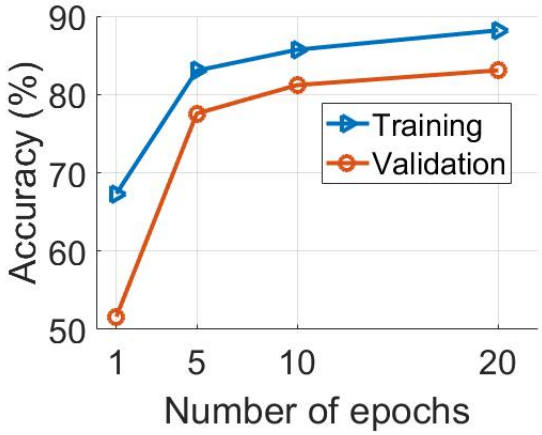

Figure S2. This plot shows $AGREL_{BTOU\_epochs\_xx}$'s training accuracy and validation accuracy on experiment 2 on day 1 for varying number of training data replications (epochs). The improvement in performance saturates roughly after 10 epochs.

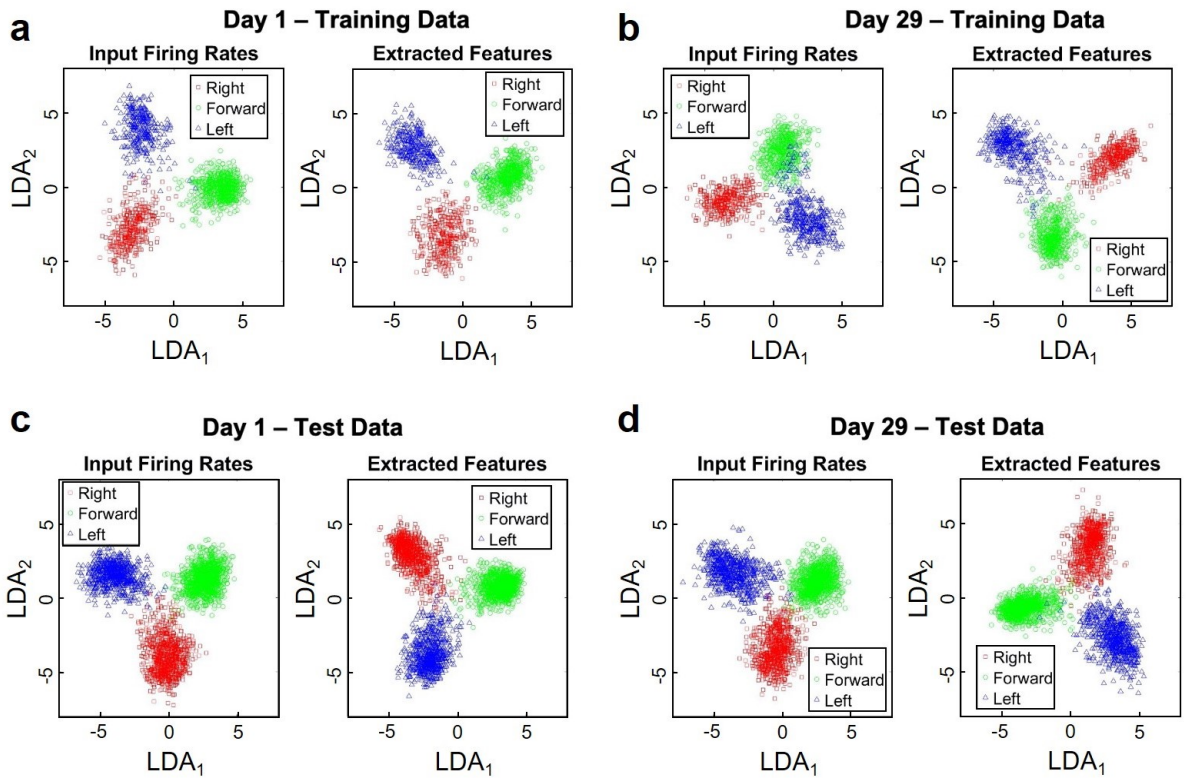

Figure S3. Low dimensional representation of input firing rates and extracted features for—**(a)** day 1 and **(b)** day 4 of training data (session 1), and **(c)** day 1 and **(d)** day 4 of test data (sessions 2 and 3) corresponding to experiment 2 dataset respectively. $AGREL_{BTOU\_transfer\_epochs\_10}$ is used to learn complex feature representations following the paradigm of transfer learning. These representations are referred to as extracted features. The separability of extracted features appears relatively better than input firing rates in the training set, whereas no improvements can be observed in the testing set. Please note that we have only shown clusters corresponding to three options instead of the original four in the experiment for ease of visualization.

References

[1] Prins NW, Sanchez JC, Prasad A. A confidence metric for using neurobiological feedback in actor-critic reinforcement learning based brain-machine interfaces. *Front. Neurosci.* 2014, 8:111.

[2] Libedinsky C, So R, Xu Z, Kyar TK, Ho D, *et al.* Independent mobility achieved through a wireless brain-machine interface. *PLoS One* 2016, 11(11):e0165773.