NeuRL: Closed-form Inverse Reinforcement Learning for Neural Decoding

Gabriel Kalweit [1 2] **Maria Kalweit** [1 2] **Mansour Alyahyay** [2 3] **Zoe Jaeckel** [2 3] **Florian Steenbergen** [2 3] **Stefanie Hardung** [2 3] **Ilka Diester** [2 3 4] **Joschka Boedecker** [1 2]

Abstract

Current neural decoding methods typically aim at explaining behavior based on neural activity via supervised learning. However, since generally there is a strong connection between learning of subjects and their expectations on long-term rewards, we hypothesize that extracting an intrinsic reward function as an intermediate step will lead to better generalization and improved decoding performance. We use inverse reinforcement learning to infer an intrinsic reward function underlying a behavior in closed form, and associate it with neural activity in an approach we call NeuRL. We study the behavior of rats in a response-preparation task and evaluate the performance of NeuRL within simulated inhibition and per-trial behavior prediction. By assigning clear functional roles to defined neuronal populations our approach offers a new interpretation tool for complex neuronal data with testable predictions. In per-trial behavior prediction, our approach furthermore improves accuracy by up to 15% compared to traditional methods.

1. Introduction

Neural decoding methods use neural spiking activity from the brain to infer predictions about behavior, like explaining or predicting movements based on activity in the motor cortex (Peixoto et al., 2021; Melbaum et al., 2021; Sani et al., 2021) or decisions based on activity located in prefrontal and parietal cortices (Baeg et al., 2003; Ibos & Freedman, 2017). Decoding can be used to control brain machine interfaces or to extract general working principles of the brain. Recently, deep learning has shown great potential in a number of domains and is outperforming classical approaches in the field of neural decoding (Glaser et al., 2019; 2017). Nevertheless, decoding methods are usually trained supervised for prediction (Xu et al., 2019; Iqbal et al., 2019), mapping greedily from neural signals directly to actions without reasoning about the long-term consequences of the actions. In the reinforcement learning (RL) paradigm, on the other hand, this is accounted for explicitly by learning a policy which maximizes long-term rewards in expectation. Prior work also showed that learning in the brain is driven by changes in the expectations about rewards and punishments (Schultz et al., 1997) which naturally aligns with the RL framework. Importantly, the immediate reward function in RL can be seen as the most succinct, robust, and transferable definition of the behavior to be learned (Abbeel & Ng, 2004). Consequently, in this work, we propose the use of *inverse reinforcement learning* (IRL) methods to infer an intrinsic reward function explaining observed animal behavior, allowing us to draw conclusions about neural activity and its relation to the recorded behaviors, as well as improving generalization and decoding performance.

We use Inverse Action-value Iteration (IAVI) (Kalweit et al., 2020) to calculate the immediate reward function analytically in closed-form assuming that a demonstrator is following a Boltzmann distribution over its unknown optimal action-values which in turn represent the expected long-term return for given taken actions. This common assumption has already been applied to model the behavior of humans and animals in a plethora of prior work (Bitterman, 1965; Feher da Silva et al., 2017; Baker et al., 2007). The learned reward function formalized in IAVI encodes the local probabilities of the demonstrated actions while enforcing the local probabilities of the maximizing actions in the future under Q-learning. In contrast, common supervised learning methods only consider the action taken in the current time step. In this work we therefore propose to instead estimate a mapping of recorded neural signals to the immediate reward function learned via IRL as an intermediate step to find coherences between neural spikings and taken actions, an approach we call *NeuRL*.

[1]Neurorobotics Lab, Department of Computer Science, University of Freiburg, Germany [2]Cluster of Excellence BrainLinks-BrainTools, University of Freiburg, Germany [3]Optophysiology Lab, Department of Biology, University of Freiburg, Germany [4]Bernstein Center Freiburg, University of Freiburg, Germany. Correspondence to: Gabriel Kalweit <kalweitg@cs.uni-freiburg.de>.

Presented at the ICML 2021 Workshop on Computational Biology (WCB).

To this end, we study the behavior of rats in a response-preparation task where rats ought to hold a lever until a cue (vibration to the paw) indicates that the animal should release. If the rats release within an allowed response window, they receive sugar water as reward. The data is recorded with electrodes spanning all cortical layers. All recorded neurons are from the Rostral Forelimb Area (RFA), which strongly contributes to planning and preparing for movements, with some having a direct connection to the Caudal Forelimb Area (CFA), responsible for motor execution.

Our contributions are threefold. First, we formalize NeuRL, a neural decoding method based on inverse action-value iteration. Second, we evaluate the performance of NeuRL within the simulated inhibition of neurons projecting from RFA to CFA and provide a comparison to real-world experimental data. Third, we evaluate NeuRL in per-trial behavior prediction showing state-of-the-art performance.

2. Background

Here we fix notation and introduce RL and IRL formally.

2.1. (Inverse) Reinforcement Learning

We model the task of neural decoding in the RL framework, where an agent (here a rat) acts in an environment as shown in Figure 1. Following policy π by applying action $a_t \sim \pi$ from n-dimensional action-space $\mathcal{A}$ in state s_t, it reaches some state $s_{t+1} \sim \mathcal{M}$ according to stochastic transition model $\mathcal{M}$ and receives scalar reward r_t in each discrete time step t. The agent has to adjust its policy π to maximize the expectation of long-term return $R(s_t) = \sum_{t'>=t} \gamma^{t'-t} r_{t'}$, where $\gamma \in [0, 1]$ is a discount factor. The action-value function then represents the expected long-term value of an action when following policy π thereupon, i.e. $Q^\pi(s_t, a_t) = \mathbf{E}_{a_{t'>t} \sim \pi, s_{t'>t} \sim \mathcal{M}}[R(s_t)|a_t]$. From the optimal action-value function Q^* one can easily derive a corresponding optimal policy π^* by maximization.

IRL recovers a reward function from observed trajectories from expert policy $\pi^\mathcal{E}$ under the assumption that the agent was (softly) maximizing the induced expected long-term return. Previous work solved this problem based on different approaches, such as e.g. Max Entropy IRL (Ziebart et al., 2008).

2.2. Action-value Iteration

We focus on the case of finding the optimal policy via model-based Action-value Iteration. The Q-function, represented by a table with entries for every state and action, gets updated in every iteration k based on the Bellman optimality equation with a given transition model $\mathcal{M}$:

$$Q_k(s_t, a_t) \leftarrow r_t + \gamma \max_a \mathbf{E}_{s_{t+1} \sim \mathcal{M}}[Q_{k-1}(s_{t+1}, a))].$$

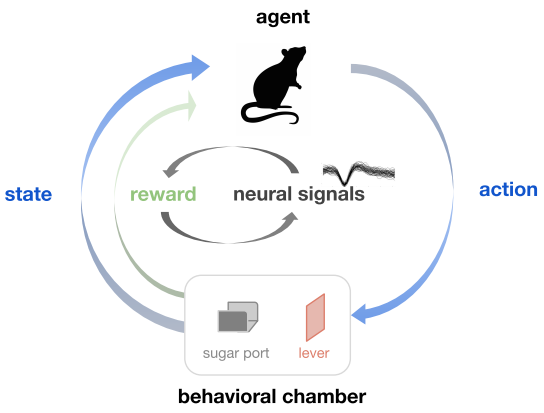

Figure 1. Response-preparation task in a reinforcement learning setting. A rat acts in a behavioral chamber with a lever and a sugar port. Our proposed framework infers an intrinsic scalar reward function of the rats behavior via closed-form inverse reinforcement learning and maps neural signals to these rewards.

3. Method

In this section, we describe how to infer the scalar underlying reward function of a rats behavior, as well as the supervised approximation of this scalar reward as a weighted combination of neural signals.

3.1. Estimation of intrinsic reward

We assume that the subject follows a stochastic policy $\pi^\mathcal{E}$ with an underlying Boltzmann distribution according to its optimal action-value function which is unknown.

As proposed in (Kalweit et al., 2020), defining:

$$\eta_s^a := \log(\pi^\mathcal{E}(a|s)) - \gamma \max_{a'} \mathbf{E}_{s' \sim \mathcal{M}}[Q^*(s', a')],$$

leads to the immediate reward:

$$r(s, a) = \eta_s^a + \frac{1}{n-1} \sum_{b \in \mathcal{A}_{\bar{a}}} r(s, b) - \eta_s^b,$$

where $\mathcal{A}_{\bar{a}} := \mathcal{A} \setminus \{a\}$. The resulting system of linear equations can be solved with least squares. We start by estimating the immediate reward for all terminal states and then go through the MDP in reverse topological order based on model $\mathcal{M}$. As can be seen in Section 5, the Boltzmann distribution induced by the optimal action-value function on this learned reward is *equivalent* to the arbitrary demonstrated behavior distribution (proof in (Kalweit et al., 2020)). IAVI thus returns a scalar reward function which precisely encodes the recorded behavior of subject rats as an intermediate result which can then be used for neural decoding.

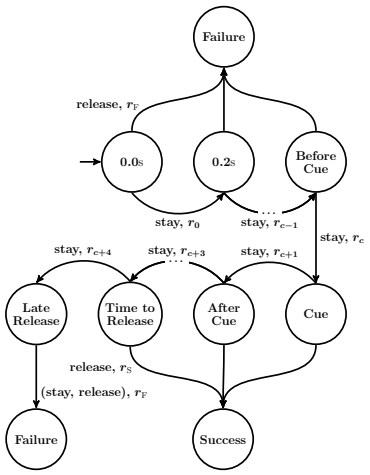

Figure 2. Transition graph for the MDP of the described response-preparation task. In the initial state, the rat presses the lever. If the rat does not release, it ends up in the next state, where the time is discretized with $0.2\,\mathrm{s}$ steps. If the rat relases after the cue in a time span of $0.6\,\mathrm{s}$, the trial was a success and it gets rewarded.

3.2. Mapping of neural spiking to intrinsic reward

As second step, we map recorded neural spikes to the found intrinsic reward function in order to draw conclusions about the recorded behavior based on neural activity. We hence assume the immediate reward function to be a projection $r(s,a) = \rho(\Phi(s)|\theta^\rho)$, where ρ is a parameterized function of features with parameters θ^ρ, e.g. a linear combination or a neural network, and $\Phi(s) = (\Phi_1(s), \dots, \Phi_m(s))^\top$ the vector of m features based on the recordings of m neurons, such as the mean. We can fit parameters θ^ρ according to the class of function approximator, e.g. either by least squares or gradient descent, on the difference between reward and prediction. The mapping can then be used to predict the resulting behavior based on neural spiking $\Phi(s)$ in new situations. Further, we can simulate how rats will behave if the spiking of certain neurons is inhibited or changed.

4. Experimental Design

In this section, we explain the experimental design.

4.1. Response-Preparation Task

A total of six rats (two for the neural recordings used in our experiments and four for the real-world inhibition experiments) were placed into a behavioral chamber with one lever and a reward port (see Figure 1). To complete the task and get the reward (sucrose water), the rats had to hold the lever for $1.6\,\mathrm{s}$ until a vibration to the paw occurs as a cue to release. The trial was considered correct if the rat released within $0.6\,\mathrm{s}$. The rats were only rewarded for correct trials and were trained for 40 sessions over the course of two months. The data set comprises recordings of 30 neurons and 104 trials of rat 1 and 33 neurons and 184 trials of rat 2.

4.2. MDP Formulation

We model a simplified version of the response-preparation task as Markov Decision Process (MDP), where we consider the task after the press of the lever. The MDP is defined as a four-tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{M}, r \rangle$, where the set of states is defined by $\mathcal{S} = \{0.0\,\mathrm{s}, 0.2\,\mathrm{s}, \dots, 1.2\,\mathrm{s}\} \cup \{\text{Before Cue}, \text{Cue}, \text{After Cue}, \text{After Cue}_1, \text{After Cue}_2, \dots, \text{Time to Release}, \text{Late Release}\} \cup \{\text{Success}, \text{Failure}\}$, discretizing the time into chunks of $0.2\,\mathrm{s}$. In every state, the rat can pick an action from action space $\mathcal{A} = \{\text{stay}, \text{release}\}$. We define the MDP to have deterministic transitions. An overview is given in Figure 2. In the following, we consider the reward function $r : \mathcal{S} \times \mathcal{A} \mapsto \mathbb{R}$ to be unknown.

4.3. Inhibition

To inhibit neurons *in vivo*, we expressed the light gated inhibitory opsin enhanced Natronomonas pharaonis Halorhodopsin (eNpHr3.0 (Gradinaru et al., 2008)) specifically targeting RFA to CFA projecting neurons in four trained rats. For this we injected a local Adeno Associated Virus (AAV)-based vector carrying the cre-dependent eNpHr construct into RFA and a retrograde traveling viral vector (retroAAV (Tervo et al., 2016)) providing cre recombinase into CFA. Thereby the opsin is only expressed in neurons projecting from RFA to CFA. Experiments were conducted 12 weeks after injection to allow high levels of opsin expression. In 25% of the trials, continuous light was delivered to RFA via optical fibers during the vibration cue.

4.4. Models

In the inhibition experiments, we compare NeuRL to real-world evidence. In the behavior prediction experiments, we compare NeuRL to a random controller, logistic regression (LR) and non-linear classification via neural-networks (NNC), which map directly from neural signal features to actions. For NeuRL and NNC, we optimized the hyperparameters with random search according to the configuration space in Table 1 with 500 sampled configurations each.

Hyperparameter	Configuration Space
#updates	[5000, **10000**, 20000]
batch size	[16, 64, **256**]
hidden dim	[**50**, 100, 200]
num layers	[$\mathbf{2}^*$, $\mathbf{3}^{**}$, 4]
learning rate	[10^{-3}, $\mathbf{10^{-4}}$, 10^{-5}]

Table 1. Configuration space for hyperparameter optimization. Incumbent in bold for (*) rat 1 and (**) rat 2.

	Rat 1			Rat 2		
	Exact Match	Near 1 Match	Near 2 Match	Exact Match	Near 1 Match	Near 2 Match
NeuRL	$\mathbf{0.36}(\pm\mathbf{0.11})$	$\mathbf{0.49}(\pm\mathbf{0.13})$	$\mathbf{0.59}(\pm\mathbf{0.09})$	$\mathbf{0.44}(\pm\mathbf{0.09})$	$\mathbf{0.62}(\pm\mathbf{0.06})$	$\mathbf{0.70}(\pm\mathbf{0.11})$
NNC	$0.21(\pm0.09)$	$0.28(\pm0.12)$	$0.37(\pm0.17)$	$0.34(\pm0.10)$	$0.46(\pm0.09)$	$0.52(\pm0.10)$
LR	$0.15(\pm0.07)$	$0.19(\pm0.10)$	$0.29(\pm0.08)$	$0.33(\pm0.09)$	$0.41(\pm0.08)$	$0.47(\pm0.10)$
Random	$0.04(\pm0.07)$	$0.20(\pm0.13)$	$0.29(\pm0.15)$	$0.12(\pm0.06)$	$0.38(\pm0.07)$	$0.46(\pm0.10)$

Table 2. Mean prediction accuracy of release time step for 10-fold cross validation on subject rat 1 and 2.

5. Experimental Results

We first learn the intrinsic reward functions based on the recorded trajectories and the above defined MDP formulation via IAVI. As can be seen in Figure 3, the learned and the real release distributions are identical, which shows that the scalar reward functions being found precisely explain the release distribution for each rat.

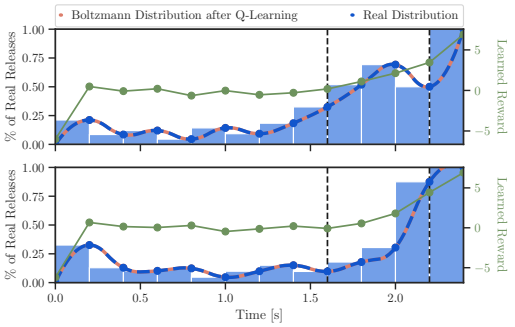

Figure 3. Release distribution, learned reward and the resulting Boltzmann distribution after applying Q-learning on the reward for (top) rat 1 and (bottom) rat 2 over all trials. Dashed lines indicate the time span in which the rats ought to release.

For the inhibition simulation, we calculate the feature matrices accumulating the neural spikings by using an incremental mean over all trials for each rat and compute the weights θ via least squares, assuming a linear combination of the state features as described in Section 3.2. We then randomly sample 60% of neurons projecting from RFA to CFA (which corresponds to the expected efficacy of viral manipulation in practice) and set the respective features within the allowed response window to zero (analogously to the real-world inhibition experiments). For each rat, we derive a stochastic policy from the Boltzmann distributions after Q-learning on the inhibition rewards induced by the modified feature matrices and sample releases from these policies. The resulting reaction times (time between cue and release in correct trials) for real and simulated inhibitions are summarized for all rats in Figure 4. The model provided by NeuRL captures the tendency towards higher reaction times found in the real-world experiments consistently for both rats. The difference in absolute numbers result from different subject rats for neural recording (basis for NeuRL) and real-world inhibition experiments.

To evaluate the performance of release behavior prediction, we split the data set into different training and test sets using 10-fold cross-validation over all trials of a rat. We take the neural spikings per time-step and trial as features to allow for per-trial prediction and use a neural network as function approximator. Since the resulting features are very sparse, we further append the time spent since trial initiation to the feature space. We compare NeuRL to NNC, LR and a random controller, considering a release prediction in a trial whenever a resulting controller assigns a probability of $> \epsilon$ (here we set $\epsilon = 0.6$) to the action of release in a certain time step. Results are shown in Table 2. NeuRL is able to correctly predict the releases in the test set by 36% and 44%, respectively, for the two rats and exceeds the performance of all baselines by a large margin, also when considering near matches within one or two time steps.

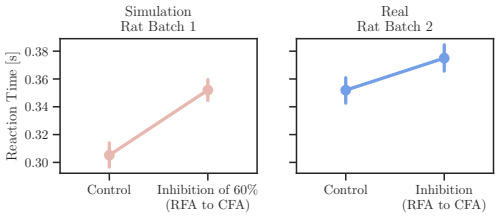

Figure 4. Mean reaction times and standard error for (left) rat batch 1 without and with simulated inhibition of 60% of RFA to CFA neurons and (right) rat batch 2 with and without real inhibition. In both cases, the reaction time increases with inhibition.

6. Conclusion

We introduced NeuRL, a two-step neural decoding method that first infers the true underlying immediate scalar reward function of a subject and then maps recorded neural spiking to this immediate reward. In simulated inhibition, our model was able to recover an effect of higher reaction times for the inhibition of neurons projecting from RFA to CFA shown in real-world experiments. In per-trial behavior prediction, our model achieved by far the best results, underlining the importance of reward prediction. Thus, our approach offers a novel and powerful interpretation tool for complex neuronal data, increasing the quality of behavioral predictions.

References

Abbeel, P. and Ng, A. Y. Apprenticeship learning via inverse reinforcement learning. In *ICML 2004*, pp. 1–. ACM, 2004. ISBN 1-58113-838-5.

Baeg, E., Kim, Y., Huh, K., Mook-Jung, I., Kim, H., and Jung, M. Dynamics of population code for working memory in the prefrontal cortex. *Neuron*, 40(1):177–188, 2003.

Baker, C., Tenenbaum, J., and Saxe, R. Goal inference as inverse planning. *Proceedings of the 29th Annual Conference of the Cognitive Science Society*, 01 2007.

Bitterman, M. E. Phyletic differences in learning. *American Psychologist*, 20(6):396, 1965.

Feher da Silva, C., Victorino, C., Caticha, N., and Baldo, M. Exploration and recency as the main proximate causes of probability matching: A reinforcement learning analysis. *Scientific Reports*, 7, 12 2017.

Glaser, J., Chowdhury, R., Perich, M., Miller, L., and Kording, K. Machine learning for neural decoding. *eneuro*, 7, 08 2017.

Glaser, J. I., Benjamin, A. S., Farhoodi, R., and Kording, K. P. The roles of supervised machine learning in systems neuroscience. *Progress in Neurobiology*, 175:126–137, 2019.

Gradinaru, V., Thompson, K. R., and Deisseroth, K. eNpHR: a Natronomonas halorhodopsin enhanced for optogenetic applications. *Brain Cell Biol*, 36(1-4):129–139, Aug 2008.

Ibos, G. and Freedman, D. J. Sequential sensory and decision processing in posterior parietal cortex. *Elife*, 6: e23743, 2017.

Iqbal, A., Dong, P., Kim, C. M., and Jang, H. Decoding neural responses in mouse visual cortex through a deep neural network. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–7, 2019.

Kalweit, G., Huegle, M., Werling, M., and Boedecker, J. Deep inverse q-learning with constraints. In *Advances in Neural Information Processing Systems*, volume 33, pp. 14291–14302. Curran Associates, Inc., 2020.

Melbaum, S., Eriksson, D., Brox, T., and Diester, I. Conserved structures of neural activity in sensorimotor cortex of freely moving rats allow cross-subject decoding. *bioRxiv*, 2021.

Peixoto, D., Verhein, J. R., Kiani, R., Kao, J. C., Nuyujukian, P., Chandrasekaran, C., Brown, J., Fong, S., Ryu, S. I., Shenoy, K. V., and Newsome, W. T. Decoding and perturbing decision states in real time. *Nature*, 591(7851): 604–609, Mar 2021.

Sani, O., Abbaspourazad, H., Wong, Y., Pesaran, B., and Shanechi, M. Modeling behaviorally relevant neural dynamics enabled by preferential subspace identification. *Nature Neuroscience*, 24:140–149, 2021.

Schultz, W., Dayan, P., and Montague, P. A neural substrate of prediction and reward. *Science*, 275:1593 – 1599, 1997.

Tervo, D. G. R., Hwang, B.-Y., Viswanathan, S., Gaj, T., Lavzin, M., Ritola, K. D., Lindo, S., Michael, S., Kuleshova, E., Ojala, D., Huang, C.-C., Gerfen, C. R., Schiller, J., Dudman, J. T., Hantman, A. W., Looger, L. L., Schaffer, D. V., and Karpova, A. Y. A designer aav variant permits efficient retrograde access to projection neurons. *Neuron*, 92(2):372–382, 2016.

Xu, Z., Wu, W., Winter, S. S., Mehlman, M. L., Butler, W. N., Simmons, C. M., Harvey, R. E., Berkowitz, L. E., Chen, Y., Taube, J. S., Wilber, A. A., and Clark, B. J. A comparison of neural decoding methods and population coding across thalamo-cortical head direction cells. *Frontiers in Neural Circuits*, 13:75, 2019.

Ziebart, B. D., Maas, A. L., Bagnell, J. A., and Dey, A. K. Maximum entropy inverse reinforcement learning. In *AAAI*, 2008.