NEURL: CLOSED-FORM INVERSE REINFORCEMENT LEARNING FOR NEURAL DECODING

Anonymous authors
Paper under double-blind review

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 propose NeuRL, an inverse reinforcement learning approach that (1) extracts an intrinsic reward function from collected trajectories of a subject in closed form, (2) maps neural signals to this intrinsic reward to account for long-term dependencies in the behavior and (3) predicts the simulated behavior for unseen neural signals by extracting Q-values and the corresponding Boltzmann policy based on the intrinsic reward values for these unseen neural signals. We show that NeuRL leads to better generalization and improved decoding performance compared to supervised approaches. 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 (Schirrmeister et al., 2017; Kuhner et al., 2019; Hübner et al., 2020) 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., 2020; 2019). 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 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 behavior, as well as improving generalization and decoding performance.

We use Inverse Action-value Iteration (IAVI) (authors, 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 observed actions. This common assumption has already been applied to model the behavior of humans and animals in a plethora of prior work (Bitterman, 1965; Baker et al., 2007; Feher da Silva et al., 2017). 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

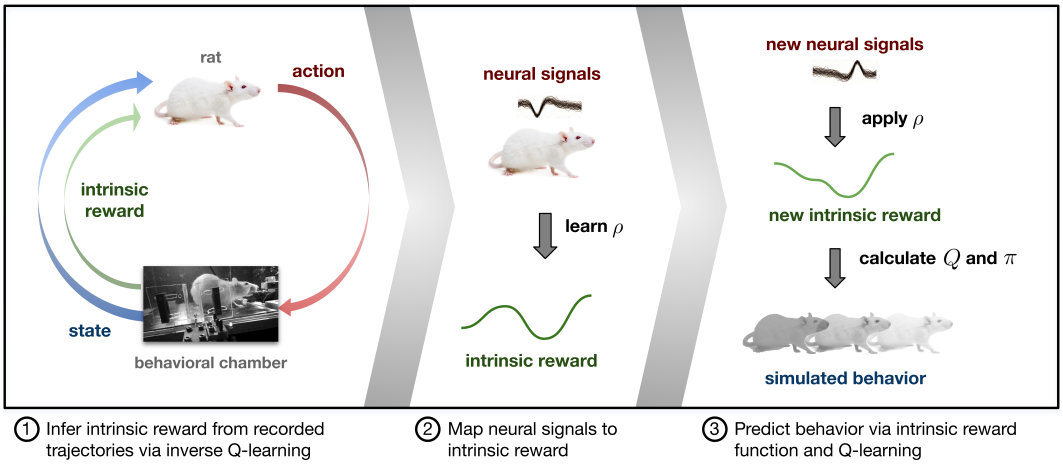

Figure 1: Response-preparation task in a reinforcement learning setting. A rat acts in a behavioral chamber with a lever and a sugar port. Our proposed framework first infers an intrinsic scalar reward function of the rat's behavior via closed-form inverse reinforcement learning. Then, a parameterized function ρ is learned which maps neural signals to the intrinsic reward. Finally, we generalize to new situations by applying ρ to other neural signals and calculating the Q-values and Boltzmann policy to study the corresponding simulated behavior for these neural signals.

under Q-learning. In contrast, common supervised learning methods only consider the action taken in the current time step. In this work we propose to instead estimate a mapping of recorded neural signals to the immediate reward function learned on observed rat trajectories via IRL as an intermediate step to find coherences between neural spikings and taken actions. The learned mapping can then be used to calculate the intrinsic reward for unseen neural signals and simulate a rat's behavior based on the new reward, an approach we call *NeuRL*. The scheme of the algorithm is shown in Figure 1. This decoding mechanism can be used to predict behavior in real-time from neural spiking or it can simulate the influence of specific neurons on the behavior of the rat. Our proposed decoding tool can help to interpret complex neural data and can serve as a hypothesis generator which then can be evaluated in vivo.

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. Figure 2 shows a rat performing this task in a behavioral chamber. 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 NeuRL in per-trial behavior prediction showing state-of-the-art performance. Third, we analyze the influence of neurons projecting from RFA to CFA by simulated inhibition within the NeuRL framework and real inhibition via viral manipulation. Our finding of similar response in real and simulated inhibition confirm that the intermediate representation of neural signals as immediate rewards offer a very promising direction for neural decoding methods.

2 Background

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 1.1. 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

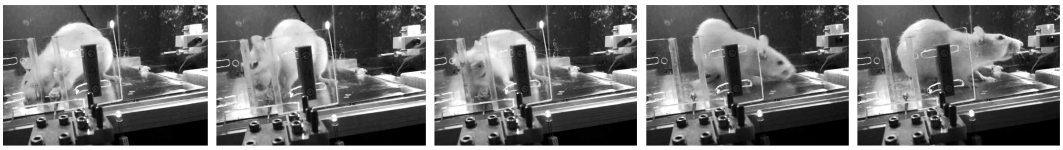

Figure 2: Successful trial of the response-preparation task in a behavioral chamber. The rat presses a lever until the vibration cue occurs, releases within $0.6\,\mathrm{s}$ and gets to the reward port.

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 $\pi^\mathcal{R}$ 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 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 $\mathcal{M}$:

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

3 Method

In this section, we describe how to infer the scalar underlying reward function of a rat's behavior, the supervised approximation of this scalar reward as a weighted combination of neural signals and the neural decoding mechanism using the intrinsic reward function.

3.1 Estimation of intrinsic reward

We assume the rodent to softly maximize its measure of optimality which we define to be the expected cumulative sum of an unknown immediate reward function, i.e. the actions taken by the animal are samples from a Boltzmann distribution over its optimal action-values $Q^*(s, \cdot)$:

$$\frac{e^{Q^*(s,a)}}{\sum_{A \in \mathcal{A}} e^{Q^*(s,A)}} \coloneqq \pi^\mathcal{R}(a|s), \tag{1}$$

for all actions $a \in \mathcal{A}$, and concomitantly:

$$e^{Q^*(s,a)} = \pi^\mathcal{R}(a|s) \sum_{A \in \mathcal{A}} e^{Q^*(s,A)} = \frac{\pi^\mathcal{R}(a|s)}{\pi^\mathcal{R}(b|s)} e^{Q^*(s,b)}, \tag{2}$$

for all actions $b \in \mathcal{A}_{\bar{a}}$ where $\mathcal{A}_{\bar{a}} = \mathcal{A} \setminus \{a\}$. Following the derivations as proposed by authors (2020):

$$Q^*(s, a) = Q^*(s, b) + \log(\pi^\mathcal{R}(a|s)) - \log(\pi^\mathcal{R}(b|s)). \tag{3}$$

Using the Bellman optimality equation in Equation (3), the immediate reward of action a in state s can be expressed by the immediate reward of some other action $b \in \mathcal{A}_{\bar{a}}$, the respective log-probabilities and future action-values:

$$\begin{aligned} r(s, a) = \log(\pi^\mathcal{R}(a|s)) - \gamma \max_{a'} \mathbf{E}_{s' \sim \mathcal{M}(s,a,s')}[Q^*(s', a')] \\ + r(s, b) - (\log(\pi^\mathcal{R}(b|s)) - \gamma \max_{b'} \mathbf{E}_{s'' \sim \mathcal{M}(s,b,s'')}[Q^*(s'', b')]). \end{aligned} \tag{4}$$

Substituting the difference between the log-probability and the discounted action-value of the future state s' as:

$$\eta_s^a \coloneqq \log(\pi^{\mathcal{R}}(a|s)) - \gamma \max_{a'} \mathbf{E}_{s' \sim \mathcal{M}(s,a,s')}[Q^*(s', a')], \tag{5}$$

we can put the reward of action a in state s in relation to the reward of all other actions:

$$r(s, a) = \eta_s^a + \frac{1}{n-1} \sum_{b \in \mathcal{A}_{\bar{a}}} r(s, b) - \eta_s^b. \tag{6}$$

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 4.3, the Boltzmann distribution induced by the optimal action-value function on this learned reward is *equivalent* to the arbitrary demonstrated behavior distribution (proof in (authors, 2020)). IAVI thus returns a scalar *intrinsic* reward function which precisely encodes the recorded behavior of subject rats as an intermediate result which can serve as supervised signal to learn a mapping from neural spiking.

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:

$$\hat{r}(\Phi(s), a) = \rho(\Phi(s)|\theta^\rho), \tag{7}$$

where ρ is a parameterized function of features with parameters θ^ρ, e.g. a linear combination or a neural network, and $\Phi(s) = (\Phi_1(s), \ldots, \Phi_m(s))^\top$ the vector of m features based on the recordings of m neurons, such as the mean activity over all trials. 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 $r(s, a)$ and prediction $\hat{r}(\Phi(s), a)$. The mapping can then be used to predict the resulting behavior based on neural spiking in new situations.

3.3 Neural Decoding From Intrinsic Reward

The parameters θ^ρ of $\hat{r}(\Phi(s), a)$ are fitted to represent immediate reward $r(s, a)$ and hence the underlying behavior of the recorded rat as closely as possible. The found parameters can contribute to generalization to any *arbitrary* neural spiking $\Psi(s) = (\Psi_1(s), \ldots, \Psi_m(s))^\top$ which yields adjusted reward and action-values in each time step t:

$$\begin{aligned} \hat{r}(\Psi(s_t), a_t) &= \rho(\hat{\Psi}(s_t)|\theta^\rho) \text{ and} \\ \hat{Q}^*(\Psi(s_t), a_t) &= \max_\pi \mathbf{E}_\pi \left[\sum_{t' \geq t} \gamma^{t'-t} \hat{r}(\Psi(s_{t'}), a_{t'}) \right]. \end{aligned} \tag{8}$$

From the optimal Q-function $Q^*(\Psi(s), a)$ based on features $\Psi(s)$, we infer the respective *predicted* action-probabilities by:

$$\hat{\pi}(a|\Psi(s)) = \frac{e^{\hat{Q}^*(\Psi(s),a)}}{\sum_{A \in \mathcal{A}} e^{\hat{Q}^*(\Psi(s),A)}}. \tag{9}$$

In order to identify neurons or groups of neurons with particular relevance for a specific type of response, we can modulate their activity by modifying the respective features $\Phi_i(s)|_{1 \leq i \leq m}$, keeping all other features fixed. Put differently, we can excite or inhibit certain neurons *within the model* and make predictions about the response. The change in behavior between the ground truth based on the recorded spiking and the predicted response based on the modulated features provide insight over the possible individual impact of these neurons on cognitive processes. Furthermore, our model offers the possibility to learn the intrinsic reward of a rat along with the respective mapping from neural spiking to rewards based on *recorded trials* in order to make predictions about behavioral response *ad hoc* in active trials.

4 EXPERIMENTS

In our experiments, we seek to find answers to the questions:

- Is the immediate reward a good intermediate representation for neural decoding?
- How does NeuRL compare to the state of the art in per-trial action prediction?
- Are the responses predicted by NeuRL in line with real-world observations in inhibition experiments?

In the following section, we first describe the response-preparation task and the respective MDP formulation. Then, we compare the action probabilities as found in the data and predicted by the controller on basis of the learned reward function by IAVI. Lastly, we employ and compare NeuRL in per-trial behavior prediction and in the context of real-world inhibition.

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 2). 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 subset of the data used for training our models 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 $\mathcal{S} = \{0.0\,\mathrm{s}, 0.2\,\mathrm{s}, \ldots, 1.2\,\mathrm{s}\} \cup \{\text{Before Cue}, \text{Cue}, \text{After Cue}, \text{After Cue}_1, \text{After Cue}_2, \ldots, \text{Time to Release}, \text{Late Release}\} \cup \{\text{Success}, \text{Failure}\}$, 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 3. In the following, we consider the reward function $r : \mathcal{S} \times \mathcal{A} \mapsto \mathbb{R}$ to be unknown.

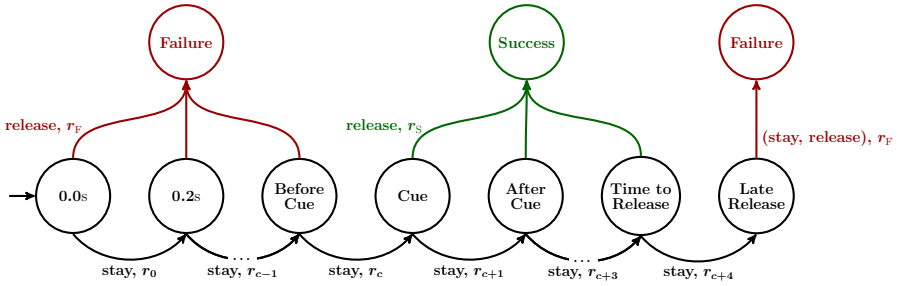

Figure 3: Transition graph for the MDP of the described response-preparation task. In the initial state, the rat presses the lever. If the rat does not release, it ends up in the next time step, where the time is discretized with $0.2\,\mathrm{s}$ steps. If the rat relases after the cue in a time span of $0.6\,\mathrm{s}$, the trial was a success and it gets rewarded. c denotes the running index over time steps before and after the cue (in our case $c = 8$, representing $1.6\,\mathrm{s}$ with $0.2\,\mathrm{s}$ steps).

4.3 REWARD ESTIMATION VIA IAVI

To verify the correctness of the immediate reward found by IAVI, we first learn the intrinsic reward functions based on the recorded trajectories of rat 1 and rat 2 and the above defined MDP formulation. As can be seen in Figure 4, the learned and real release distributions are identical, which shows that the scalar reward functions being found precisely explain the release distribution for each rat.

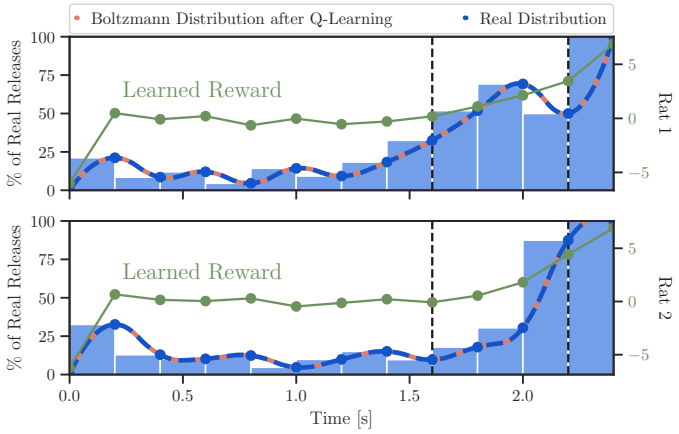

Figure 4: Release distribution, learned reward and the resulting Boltzmann distribution after applying Q-learning on the reward for (top) rat 1 and (bottom) rat 2 over all trials. Dashed lines indicate the time span in which the rats ought to release.

4.4 Per-Trial Behavior Prediction from Neural Spiking

To study the performance for predicting actions of rats based on their neural signals in a trial with NeuRL, we use the neural spikings per time-step and trial as features and a neural network as function approximator for the reward. Since the resulting features are very sparse, we further append the time spent since trial initiation to the feature space. The learned intrinsic reward function is used to compute the corresponding release policy by applying action-value iteration on the reward. 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. 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, we consider it a predicted release. We split the data set introduced in Section 4.1 into different training and test sets using 10-fold cross-validation over all trials of a rat. 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	[**2***, **3****, 4]
learning rate	[10^{-3}, **10^{-4}**, 10^{-5}]

(a) Incumbents of NeuRL.

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

(b) Incumbents of NNC.

Table 1: Configuration space of hyperparameters. **Incumbent** for rat 1 (*) and rat 2 (**).

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. An intuition of why the immediate reward is a good intermediate representation can be gained from the visualization of the latent representation of the last hidden layers for the classifier (NNC) and NeuRL in Figure 5. As substitute for a Q-value, we show the normalized cumulative embedding of the immediate reward. The latent representation of the neural features grounded in the learned immediate reward preserves the temporal coherence which stands in contrast to the latent embedding of the classifier. This brings light to the advantages of the proposed representation. The larger increase in correct releases near two time steps, as shown in Table 3, further strengthens this point since most overlap in the latent embedding of NeuRL is between similar time steps (cf. Figure 5).

	Rat 1		
	Exact Match	Near 1 Match	Near 2 Match
NeuRL	$\mathbf{0.36}(\pm\mathbf{0.11})$	$\mathbf{0.49}(\pm\mathbf{0.13})$	$\mathbf{0.59}(\pm\mathbf{0.09})$
NNC	$0.21(\pm0.09)$	$0.28(\pm0.12)$	$0.37(\pm0.17)$
LR	$0.15(\pm0.07)$	$0.19(\pm0.10)$	$0.29(\pm0.08)$
Random	$0.04(\pm0.07)$	$0.20(\pm0.13)$	$0.29(\pm0.15)$

	Rat 2		
	Exact Match	Near 1 Match	Near 2 Match
NeuRL	$\mathbf{0.44}(\pm\mathbf{0.09})$	$\mathbf{0.62}(\pm\mathbf{0.06})$	$\mathbf{0.70}(\pm\mathbf{0.11})$
NNC	$0.34(\pm0.10)$	$0.46(\pm0.09)$	$0.52(\pm0.10)$
LR	$0.33(\pm0.09)$	$0.41(\pm0.08)$	$0.47(\pm0.10)$
Random	$0.12(\pm0.06)$	$0.38(\pm0.07)$	$0.46(\pm0.10)$

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

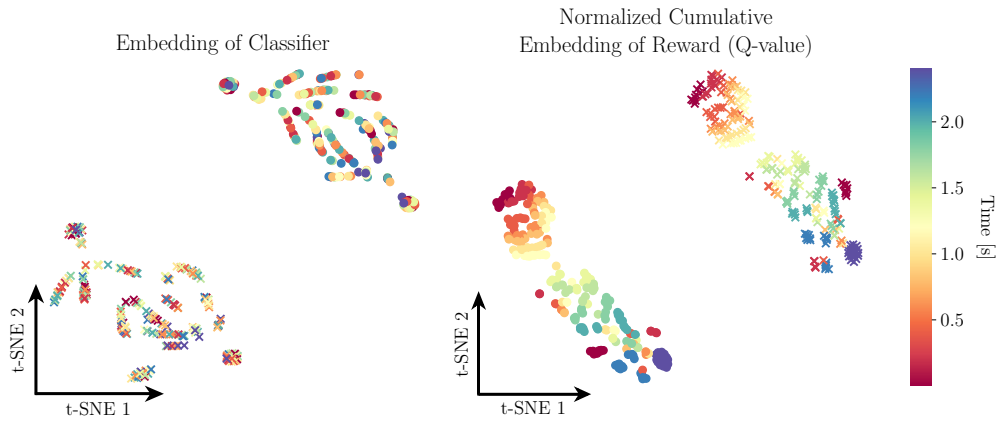

Figure 5: (left) Visualization of latent embeddings for the two actions *stay* ($\circ$) and *release* ($\times$) generated from the last hidden layer of the classifier (NNC). (right) Visualization of the normalized cumulative latent embeddings generated by the reward-model (as substitute for a Q-value). Our model preserves the temporal coherence of the task for the two actions *stay* ($\circ$) and *release* ($\times$) much better than the classifier on the left which is necessary for correct release prediction.

	Rat 1	Rat 2
NeuRL	$+\mathbf{23}\%$	$+\mathbf{26}\%$
NNC	$+16\%$	$+18\%$

Table 3: Increase in mean prediction accuracy for rat 1 and 2 going from exact to near 2 matches.

4.5 Simulation of Neural Inhibition and its Effect on the Rat's Behavior

We study the influence of neurons projecting from RFA to CFA (cf. Figure 6A) of which we identified 10 using optogenetic phototagging. The temporal pattern of the firing rate, as depicted in Figure 6B, is surprisingly diverse even for a specific pathway. While the majority of the neurons (6/10, 60%) increase the firing rate in the response period (indicated by dashed lines), about one third of the neurons (3/10, 30%) have a higher firing rate in the hold period. Since most of the neurons are more active in the response window, we hypothesize that inhibition during this period has a significant effect on motor execution.

First, we simulate the effect of inhibition of these neurons within NeuRL and use the recordings of rats without viral manipulation. To simulate varying expected efficacy of viral manipulation, we sample subsets of the neurons projecting from RFA to CFA and set the respective features in $\Psi(s)$ within the allowed response window to zero (analogously to real-world inhibition experiments). We

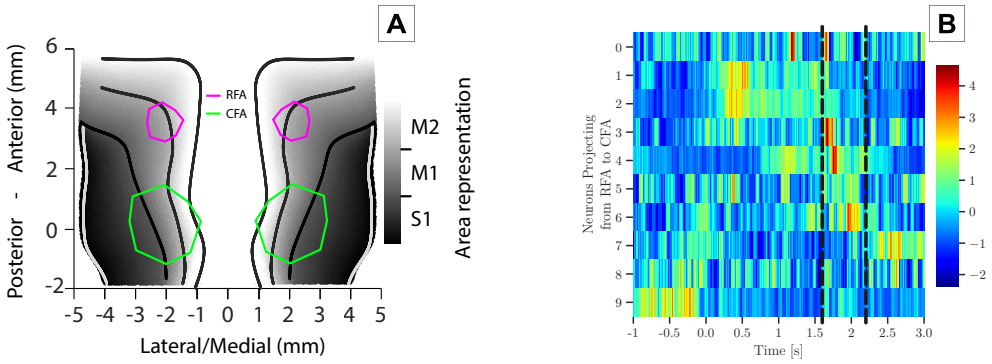

Figure 6: **A** Delineation of the Rostral (RFA) and Caudal (CFA) Forelimb Areas in a rat's brain according to Neafsey & Sievert (1982) and Rouiller et al. (1993). **B** Z-score normalized firing rates of neurons projecting from RFA to CFA.

calculate the feature matrices accumulating the neural spikings by using an incremental mean over all trials for each rat (to aggregate all available information) and compute the weights θ^{ρ} via least squares, assuming a linear combination of the state features as described in Section 3.2. Then, as described above, we map the features to intrinsic rewards and compute the Q-values and corresponding stochastic Boltzmann policies.

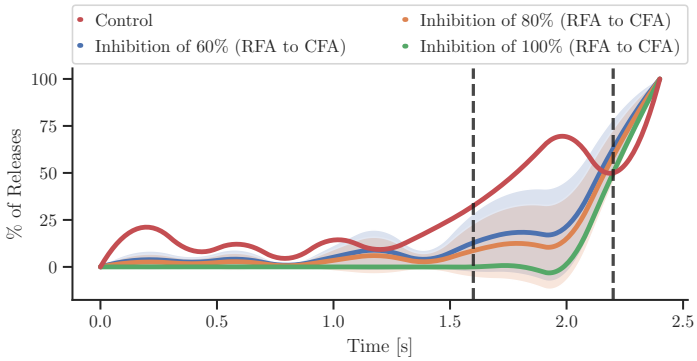

Figure 7: Release distribution of the rats (Control) and the resulting Boltzmann policies after applying Q-learning on the modified reward for different levels of simulated inhibition of neurons projecting from RFA to CFA. Dashed lines indicate the time span in which the rats ought to release.

Figure 7 shows the resulting release probabilities for different shares of inhibited neurons $(0.0, 0.6, 0.8$ and $1.0)$ in the response window between $1.6\,\mathrm{s}$ and $2.2\,\mathrm{s}$. The inhibition causes late releases, as the probability of correct releases between $1.6\,\mathrm{s}$ and $2.2\,\mathrm{s}$ decreases with a higher proportion of inhibited neurons. To evaluate our model, we consider trajectories of rodents solving the response-preparation task as defined in Section 4.1 after neural inhibition via viral manipulation. 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.

In order to further put the performance of NeuRL in context of the current state of the art (Glaser et al., 2020), we compare to logistic regression and NNC as described in Section 4.4. We train the baseline models on the recorded trials without viral manipulation on the basis of single time steps

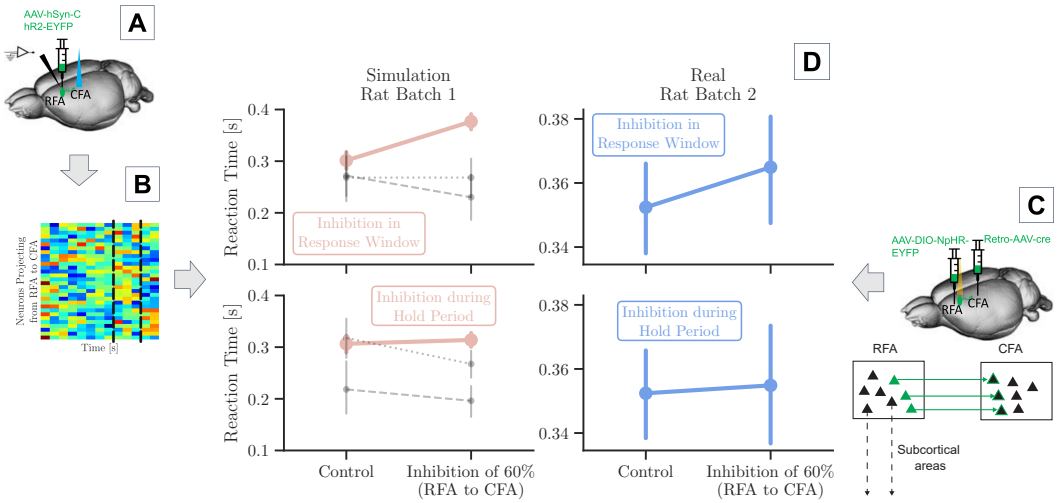

Figure 8: A Identification of relevant neurons via optogenetic phototagging. B Processing of neural recordings and extraction of neural spiking. C Viral manipulation of the pathway projecting from RFA to CFA. D Mean reaction times and standard error for (left) rat batch 1 without and with simulated inhibition of 60% of RFA to CFA neurons and (right) rat batch 2 with and without real inhibition. (top) Within NeuRL (red) and in the real-world experiments (blue), the reaction time increases with inhibition in the response window. (bottom) There is no significant change in reaction time with inhibition during the hold period. Baselines are depicted in black for comparison, (- -) for linear regression and (····) for NNC.

(analogously to our experiments in Section 4.4, the time spent since trial initiation is added to the feature space to account for the sparsity of per-time step features) and then use the predicted release probabilities according to the modified features $\Psi(s)$.

The resulting reaction times (time between cue and release in correct trials) for real and simulated inhibition with an efficacy of 60% of the neurons projecting from RFA to CFA (corresponding to the efficacy of viral manipulation in practice) are summarized for all rats in Figure 8. The model provided by NeuRL captures both the tendency towards higher reaction times found in the real-world experiments of viral manipulation in the response window, as well as the absence of delay for inhibition during the hold period 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. In contrast, logistic regression and non-linear classification are not able to reproduce the findings found in the in vivo experiments.

5 Conclusion

We introduced NeuRL, a three-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 order to provide the possibility to decode unseen neural recordings thereafter. 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.

REPRODUCIBILITY STATEMENT

Data recordings will be made available upon release of a later journal submission. To this point, we release a working pipeline for application. Hyperparameters are listed in the paper.

REFERENCES

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

Anonymous authors. Deep inverse q-learning with constraints. In *Advances in Neural Information Processing Systems*, volume 33, pp. 14291–14302. Curran Associates, Inc., 2020.

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

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

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

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

Joshua I. Glaser, Ari S. Benjamin, Roozbeh Farhoodi, and Konrad P. Kording. The roles of supervised machine learning in systems neuroscience. *Progress in Neurobiology*, 175:126–137, 2019.

Joshua I Glaser, Ari S Benjamin, Raeed H Chowdhury, Matthew G Perich, Lee E Miller, and Konrad P Kording. Machine learning for neural decoding. *Eneuro*, 7(4), 2020.

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

David Hübner, Albrecht Schall, and Michael Tangermann. Unsupervised learning in a bci chess application using label proportions and expectation-maximization. *Brain-Computer Interfaces*, 7(1-2):22–35, 2020. doi: 10.1080/2326263X.2020.1741072. URL https://doi.org/10.1080/2326263X.2020.1741072.

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

Asim Iqbal, Phil Dong, Christopher M Kim, and Heeun Jang. 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.

D. Kuhner, L.D.J. Fiederer, J. Aldinger, F. Burget, M. Völker, R.T. Schirrmeister, C. Do, J. Boedecker, B. Nebel, T. Ball, and W. Burgard. A service assistant combining autonomous robotics, flexible goal formulation, and deep-learning-based brain–computer interfacing. *Robotics and Autonomous Systems*, 116:98–113, 2019. ISSN 0921-8890. doi: https://doi.org/10.1016/j.robot.2019.02.015. URL https://www.sciencedirect.com/science/article/pii/S0921889018302227.

Svenja Melbaum, David Eriksson, Thomas Brox, and Ilka Diester. Conserved structures of neural activity in sensorimotor cortex of freely moving rats allow cross-subject decoding. *bioRxiv*, 2021.

E.J. Neafsey and Carl Sievert. A second forelimb motor area exists in rat frontal cortex. *Brain Research*, 232(1):151–156, 1982. ISSN 0006-8993. doi: https://doi.org/10.1016/0006-8993(82)90617-5. URL https://www.sciencedirect.com/science/article/pii/0006899382906175.

Diogo Peixoto, Jessica R. Verhein, Roozbeh Kiani, Jonathan C. Kao, Paul Nuyujukian, Chandramouli Chandrasekaran, Julian Brown, Sania Fong, Stephen I. Ryu, Krishna V. Shenoy, and William T. Newsome. Decoding and perturbing decision states in real time. *Nature*, 591(7851): 604–609, Mar 2021.

Eric M. Rouiller, Veronique Moret, and Fengyi Liang. Comparison of the connectional properties of the two forelimb areas of the rat sensorimotor cortex: Support for the presence of a premotor or supplementary motor cortical area. *Somatosensory & Motor Research*, 10(3): 269–289, 1993. doi: 10.3109/08990229309028837. URL https://doi.org/10.3109/08990229309028837.

Omid G. Sani, Hamidreza Abbaspourazad, Yan T. Wong, Bijan Pesaran, and Maryam M. Shanechi. Modeling behaviorally relevant neural dynamics enabled by preferential subspace identification. *Nature Neuroscience*, 24:140–149, 2021.

Robin Tibor Schirrmeister, Jost Tobias Springenberg, Lukas Dominique Josef Fiederer, Martin Glasstetter, Katharina Eggensperger, Michael Tangermann, Frank Hutter, Wolfram Burgard, and Tonio Ball. Deep learning with convolutional neural networks for eeg decoding and visualization. *Human Brain Mapping*, 38(11):5391–5420, 2017. doi: https://doi.org/10.1002/hbm.23730. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/hbm.23730.

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

D. Gowanlock R. Tervo, Bum-Yeol Hwang, Sarada Viswanathan, Thomas Gaj, Maria Lavzin, Kimberly D. Ritola, Sarah Lindo, Susan Michael, Elena Kuleshova, David Ojala, Cheng-Chiu Huang, Charles R. Gerfen, Jackie Schiller, Joshua T. Dudman, Adam W. Hantman, Loren L. Looger, David V. Schaffer, and Alla Y. Karpova. A designer aav variant permits efficient retrograde access to projection neurons. *Neuron*, 92(2):372–382, 2016.

Zishen Xu, Wei Wu, Shawn S. Winter, Max L. Mehlman, William N. Butler, Christine M. Simmons, Ryan E. Harvey, Laura E. Berkowitz, Yang Chen, Jeffrey S. Taube, Aaron A. Wilber, and Benjamin J. Clark. A comparison of neural decoding methods and population coding across thalamo-cortical head direction cells. *Frontiers in Neural Circuits*, 13:75, 2019.

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