LETTER Communicated by Christopher DiMattina

Active Learning for Discrete Latent Variable Models

Aditi Jha
aditijha@princeton.edu
Princeton Neuroscience Institute and Department of Electrical and Computer Engineering, Princeton University, Princeton, NJ 08544, U.S.A.

Zoe C. Ashwood
zashwood@princeton.edu
Princeton Neuroscience Institute and Department of Computer Science, Princeton University, Princeton, NJ 08544, U.S.A.

Jonathan W. Pillow
pillow@princeton.edu
Princeton Neuroscience Institute, Princeton University, Princeton, NJ 08544, U.S.A.

Active learning seeks to reduce the amount of data required to fit the parameters of a model, thus forming an important class of techniques in modern machine learning. However, past work on active learning has largely overlooked latent variable models, which play a vital role in neuroscience, psychology, and a variety of other engineering and scientific disciplines. Here we address this gap by proposing a novel framework for maximum-mutual-information input selection for discrete latent variable regression models. We first apply our method to a class of models known as mixtures of linear regressions (MLR). While it is well known that active learning confers no advantage for linear-gaussian regression models, we use Fisher information to show analytically that active learning can nevertheless achieve large gains for mixtures of such models, and we validate this improvement using both simulations and real-world data. We then consider a powerful class of temporally structured latent variable models given by a hidden Markov model (HMM) with generalized linear model (GLM) observations, which has recently been used to identify discrete states from animal decision-making data. We show that our method substantially reduces the amount of data needed to fit GLM-HMMs and outperforms a variety of approximate methods based on variational and amortized inference. Infomax learning for latent variable models thus offers a powerful approach for characterizing temporally structured latent states, with a wide variety of applications in neuroscience and beyond.

Neural Computation **36, 437–474** (2024)
https://doi.org/10.1162/neco_a_01646

1 Introduction

Obtaining labeled data is a key challenge in many scientific and machine learning applications. Active learning provides a solution to this problem, allowing researchers to identify the most informative data points and thereby minimize the number of examples needed to fit a model. Bayesian active learning, also known as optimal or adaptive experimental design (Verdinelli & Kadane, 1992; Chaloner & Verdinelli, 1995; Cohn et al., 1996; Ryan et al., 2016), has had a major impact on a variety of disciplines, including neuroscience (Lewi et al., 2007; Lewi et al., 2009; Lewi et al., 2011; DiMattina & Zhang, 2011; Gollisch & Herz, 2012; Shababo et al., 2013; DiMattina & Zhang, 2013; Kim et al., 2014; Park et al., 2014; Pillow & Park, 2016), psychology (Watson & Pelli, 1983; Myung et al., 2013; DiMattina, 2015; Watson, 2017; Bak & Pillow, 2018), genomics (Steinke et al., 2007), and compressed sensing (Seeger, 2008; Seeger & Nickisch, 2008; Vasisht et al., 2014).

The general setting for Bayesian active learning involves a probabilistic model $P(y \mid \mathbf{x}, \theta)$, in which a parameter vector θ governs the probabilistic relationship between inputs $\mathbf{x}$ and labels or outputs y. To improve learning of θ, we wish to select inputs $\{\mathbf{x}_t\}$ that will allow us to best estimate θ from the resulting data set $\{\mathbf{x}_i, y_i\}_{i=1}^{t}$. In standard fixed-design experiments, the inputs are selected in advance or drawn randomly from a predetermined distribution. In adaptive or closed-loop experiments, by contrast, the inputs are selected adaptively during the experiment based on the measurements obtained so far. Bayesian active learning methods provide a framework for optimally selecting these inputs, where optimality is defined by a utility function that characterizes the specific learning objective (MacKay, 1992; Cohn et al., 1996; Roy & McCallum, 2001; Pillow & Park, 2016).

Despite a burgeoning literature, the active learning field has devoted relatively little attention to latent variable models (but see Cohn et al., 1996; Hefang et al., 2000; and Anderson & Moore, 2005). Latent variable models (LVMs) represent a class of highly expressive models with a vast range of applications. In neuroscience, they have provided powerful descriptions of both neural population activity (Rainer & Miller, 2000; Kemere et al., 2008; Miller & Katz, 2010; Yu et al., 2009; Chen et al., 2009; Escola et al., 2011; Linderman et al., 2016; Glaser et al., 2020; Zoltowski et al., 2020; Jha et al., 2021) and animal behavior (Wiltschko et al., 2015; Calhoun et al., 2019; Ashwood et al., 2022; Bolkan et al., 2022; Weilnhammer et al., 2021; Zucchini et al., 2008).

A particular class of LVMs, namely hidden Markov models with generalized linear model observations (GLM-HMMs), has recently been used to identify internal states from animal behavior during decision making (Ashwood et al., 2022; Bolkan et al., 2022; Yin et al., 2023). The latent states in GLM-HMMs allow them to capture multiple behavioral strategies that an animal uses while performing a decision-making task. However, these models require large amounts of data to obtain accurate fits. In past work,

GLM-HMMs have only been applied to data sets containing many sessions collected across multiple days, weeks, or months. This heavy data requirement motivated us to develop an active learning method for GLM-HMMs as well as more general classes of discrete LVMs. Active learning, in such a setting, can allow us to adaptively select the stimulus that will reveal the most information about the animal's full set of decision-making strategies and thus reduce the number of trials needed to characterize the internal states underlying its behavior.

The key feature of latent-variable-based regression models is the relationship between input $\mathbf{x}$ and output y, mediated by an unobserved or hidden state variable z. This provides such models with the flexibility to describe internal states of the system that cannot be observed directly. However, this flexibility comes with a cost: the likelihood (and, by extension, the posterior) in LVMs is usually not available in closed form. This complicates posterior inference and the calculation of expected utility, both required for Bayesian active learning algorithms.

To address this gap in the literature, we introduce a Bayesian active learning framework for discrete latent variable models.[1] We develop methods based on both Markov chain Monte Carlo (MCMC) sampling and variational inference to efficiently compute information gain and select informative inputs in adaptive experiments. We illustrate our framework with applications to two specific families of latent variable models: (1) a mixture of linear regressions (MLR) model and (2) input-output hidden Markov models with generalized linear model (GLM) observations (GLM-HMM). We compare the efficiency of different methods, including a recent method based on amortized inference using deep networks (Foster et al., 2021), and show that in both model families, our approach provides dramatic speedups in learning model parameters over previous methods.

2 Related Work

Bayesian active learning methods have been developed for a wide range of different models, from generalized linear models (Chaloner, 1984; Paninski, 2005; Khuri et al., 2006; Lewi et al., 2007, 2009, 2011; Bak et al., 2016; Bak & Pillow, 2018) to neural networks (Cohn et al., 1996; DiMattina & Zhang, 2011, 2013; Cowley et al., 2017; Gal et al., 2017; Kirsch et al., 2019; Wu et al., 2021).

One body of work has focused on Bayesian active learning for models with implicit likelihoods (Kleinegesse & Gutmann, 2020; Ivanova et al., 2021). Another recent line of work has focused on general-purpose

[1] Code available at https://github.com/97aditi/active_learning_latent_variable_models.

real-time active learning using amortized inference in deep neural networks, an approach known as deep adaptive design (DAD; Foster et al., 2021). However, the literature on active learning for latent variable models is sparse, limited to a few specific model classes and tasks such as density modeling (Cohn et al., 1996; Hefang et al., 2000) and state estimation for standard HMMs (Anderson & Moore, 2005). The approach we develop here grows out of previous work on Bayesian active learning methods for generalized linear models (Lewi et al., 2007, 2009; Houlsby et al., 2011; Bak & Pillow, 2018). Our primary contribution is to extend these methods to discrete latent variable models, especially those used in neuroscience.

3 Discrete Latent Variable Models (LVMs)

Before turning to the problem of active learning, we provide a brief description of discrete latent variable regression models. The model has two basic components: a prior over the latent variable and a conditional distribution of the response given the stimulus and latent. Formally, this model architecture can be expressed by a pair of equations,

$$z \sim P(z \mid \theta), \tag{3.1}$$

$$y \mid \mathbf{x}, z \sim P(y \mid \mathbf{x}, z, \theta), \tag{3.2}$$

where $z \in \{1, \ldots, K\}$ is a discrete latent variable governing the internal state of the system, $\mathbf{x} \in \mathbb{R}^D$ is the input or stimulus, $y \in \mathcal{Y}$ is the response (which may be continuous or discrete), and $\theta \in \Omega$ denotes a set of model parameters governing both prior and conditional response distributions. Figure 1A shows an illustration of an example discrete latent variable model, where the conditional distribution of the response given the stimulus and latent is given by a generalized linear model. The discrete latent variable z governs which of the three generalized linear models determines the response for a given trial.

They key difficulty in working with latent variable models is that the conditional probability of the response given the stimulus requires marginalizing over the latent variable. This conditional distribution is given by

$$P(y \mid \mathbf{x}, \theta) = \sum_{k=1}^{K} P(y \mid \mathbf{x}, z = k, \theta) P(z = k \mid \theta), \tag{3.3}$$

which involves a numerical sum over all possible values of the latent state. In models where the latent variable exhibits additional structure, such as hidden Markov models (HMMs), computing this sum relies on specialized algorithms, which we discuss in more detail below.

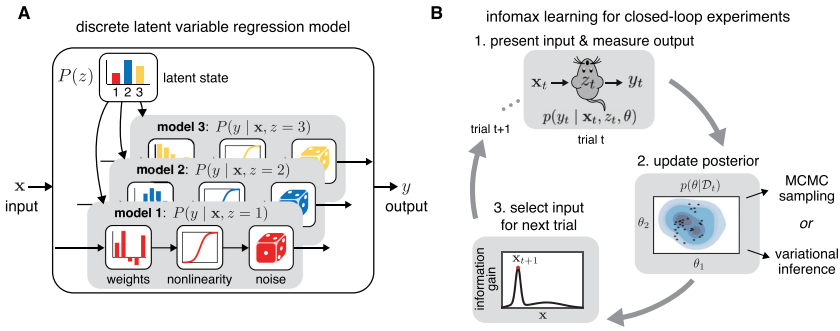

Figure 1. Discrete latent variable regression models and infomax learning. (A) Schematic of a discrete latent variable model for regression settings. The response y of the model given a stimulus $\mathbf{x}$ and a latent z is produced by generalized linear models. Here the discrete latent variable z determines which of the three generalized linear models at the bottom determines the input-output mapping on any trial. (B) Infomax learning for discrete latent variable models. On trial t, present an input $\mathbf{x}_t$ to the system of interest (e.g., a mouse performing a decision-making task) and record its response y_t. We assume this response depends on the stimulus (input), as well as an internal or latent state z_t, as specified by the model $P(y_t \mid \mathbf{x}_t, z_t, \theta)$. Second, update the posterior distribution over model parameters θ given the data collected so far in the experiment, $\mathcal{D}_t = \{\mathbf{x}_{1:t}, y_{1:t}\}$, using either MCMC sampling or variational inference. Third, select the input for the next trial that maximizes information gain or the mutual information between the next response y_{t+1} and the model parameters θ.

4 Infomax Learning

The general goal of active learning is to select inputs that will allow us to infer the model parameters θ using as few samples as possible. Bayesian active learning formalizes this in terms of a utility function that specifies the goal of learning, for example, to maximize mutual information (Lewi et al., 2009), minimize mean-squared error (Kuck et al., 2006), or minimize prediction error (Cohn et al., 1996; Roy & McCallum, 2001).

Here we select mutual information as our utility function, specifically the mutual information between response y and the model parameters θ, conditioned on the input $\mathbf{x}$. Intuitively, this rule corresponds to selecting the stimulus for which the resulting response will provide the greatest reduction in uncertainty about the model parameters, quantified in bits. Active learning with mutual information as utility is commonly known as *infomax learning*, and it has been widely applied in both machine learning and neuroscience settings (MacKay, 1992; Lewi et al., 2007; Lewi et al., 2009; Park et al., 2014; Houlsby et al., 2011; Pillow & Park, 2016; Bak & Pillow, 2018; DiMattina & Zhang, 2011).

Typical frameworks for infomax learning involve a "greedy" approach, where inputs are selected one at a time to maximize information provided by y about θ on each trial. In this setting, the experimenter selects the stimulus $\mathbf{x}_t$ on trial t according to

$$\mathbf{x}_t = \arg\max_{\mathbf{x}} I(\theta, y \mid \mathbf{x}, \mathcal{D}_{t-1}), \tag{4.1}$$

where I represents mutual information, y is the (as yet unobserved) response on trial t, while also conditioning on $\mathcal{D}_{t-1} = \{(\mathbf{x}_\tau, y_\tau)\}_{\tau=1}^{t-1}$, the stimulus-response data collected previously in the experiment. This selection rule is equivalent to saying that we maximize the expected information gain about θ or minimize the expected entropy of the posterior over θ (MacKay, 1992).

The mutual information (also known as Shannon information) between y and θ given $\mathbf{x}$ and $\mathcal{D}_{t-1}$ can be written in several equivalent forms (Cover & Thomas, 1991), one of which is

$$I(\theta\ ; y \mid \mathbf{x}, \mathcal{D}_{t-1}) = H(y\ ;\ \mathbf{x}, \mathcal{D}_{t-1}) - H(y \mid \theta\ ;\ \mathbf{x}, \mathcal{D}_{t-1}), \tag{4.2}$$

where the second term denotes the conditional entropy of y given θ, given by

$$H(y \mid \theta\ ;\ \mathbf{x}, \mathcal{D}_{t-1}) = -\int_\Omega \int_\mathcal{Y} P(y, \theta \mid \mathbf{x}, \mathcal{D}_{t-1}) \log P(y \mid \theta, \mathbf{x}, \mathcal{D}_{t-1})\, dy\, d\theta, \tag{4.3}$$

and the first term is the marginal entropy of y:

$$H(y\ ;\ \mathbf{x}, \mathcal{D}_{t-1}) = -\int_\mathcal{Y} P(y \mid \mathbf{x}, \mathcal{D}_{t-1}) \log P(y \mid \mathbf{x}, \mathcal{D}_{t-1})\, dy, \tag{4.4}$$

where both terms are conditioned on the stimulus $\mathbf{x}$ and previously collected data $\mathcal{D}_{t-1}$. In the above expressions, the integrals over y can be replaced by sums when y is discrete.

Note that for the entropy terms defined above, we use a semicolon to denote a quantity that is conditioned on a particular value of a random variable, and a vertical line to denote a conditional entropy, which by definition requires marginalizing over the random variable in question. Hence, $H(y \mid \theta\ ;\ \mathbf{x}, \mathcal{D}_{t-1})$ denotes the conditional entropy of y given θ (which involves an integral over θ) while conditioning on the specific input $\mathbf{x}$ and data from all previous trials $\mathcal{D}_{t-1}$.

5 Infomax Learning for Discrete LVMs

The challenge in applying infomax learning to latent variable models is that the posterior over the model parameters, $P(\theta \mid \mathcal{D}_{t-1})$, as well as the conditional response distribution, $P(y \mid \theta, \mathbf{x}, \mathcal{D}_{t-1})$, are not available in closed form due to the fact that they require marginalization over the latent variable. In fact, for discrete latent variable models, these distributions are not even guaranteed to be unimodal (unlike in generalized linear models, for example). Furthermore, the marginal response distribution $P(y \mid \mathbf{x}, \mathcal{D}_{t-1})$ (in equation 4.4), requires marginalizing the conditional response distribution over the parameters,

$$P(y \mid \mathbf{x}, \mathcal{D}_{t-1}) = \int P(y \mid \theta, \mathbf{x}, \mathcal{D}_{t-1})\, P(\theta \mid \mathcal{D}_{t-1})\, d\theta, \tag{5.1}$$

which exacerbates the problem of rapidly computing and optimizing the mutual information between trials.

To overcome this challenge, we develop two different approaches for infomax active learning in discrete latent variable models: one based on sampling (Houlsby et al., 2011; Bak & Pillow, 2018) and another based on variational inference (VI) (Blei et al., 2017), which we describe in the next two sections.

Figure 1B shows an illustration of infomax active learning for discrete LVMs in the context of a neuroscience experiment. On trial t, the animal receives an input $\mathbf{x}_t$ and generates a response y_t. We then update the posterior distribution over θ given all previous data using either an MCMC-sampling-based method (see section 5.1) or a variational inference method (see section 5.2). We use samples from these posteriors to evaluate the expectations required for computing mutual information and select the stimulus $\mathbf{x}$ for trial $t+1$ that maximizes $I(y_{t+1}, \theta \mid \mathbf{x}, \mathcal{D}_t)$, the conditional mutual information between the response and model parameters.

5.1 MCMC Sampling-Based Method. First, we propose a method for infomax learning of discrete latent variable models that relies on Markov chain Monte Carlo (MCMC) sampling. Specifically, we use Gibbs sampling to draw samples of θ from $P(\theta \mid \mathcal{D}_{t-1})$, the posterior distribution over parameters given the data collected so far in the experiment. These samples are then used to evaluate the conditional mutual information gain, as described below.

Gibbs sampling allows us to obtain an alternating chain of samples of the latents $z_{1:t-1}$ and the model parameter θ from their joint conditional distribution $P(\theta, z_{1:t-1} \mid \mathcal{D}_{t-1})$. We then discard the latent samples and keep only the samples of θ, $\{\theta^j\}_{j=1}^M \sim P(\theta \mid \mathcal{D}_{t-1})$, for some number of samples M, thus marginalizing over the latents. This, however, is not trivial for models

where the conditional $P(\theta \mid z_{1:t-1}, \mathcal{D}_{t-1})$ is not available in closed form (such as GLM-HMMs). We developed a modified version of Gibbs sampling for such cases, which we discuss in detail in section 7.

Each sample θ^j parameterizes a model with conditional probability of the response y given by $P(y \mid \theta^j, \mathbf{x}, \mathcal{D}_{t-1})$, which can be evaluated by marginalizing over the discrete latents (using equation 3.3). This allows us to compute the marginal probability of the response y conditioned on the stimulus and past data:

$$P(y \mid \mathbf{x}, \mathcal{D}_{t-1}) \approx \frac{1}{M} \sum_{j=1}^{M} P(y \mid \theta^j, \mathbf{x}, \mathcal{D}_{t-1}). \tag{5.2}$$

Using these conditional and marginal response probabilities, we can next compute sample-based versions of the entropy terms (see equations 4.3 and 4.4) as follows:

$$H(y \mid \theta\ ;\ \mathbf{x}, \mathcal{D}_{t-1}) \approx \frac{1}{M} \sum_{j=1}^{M} \int_{\mathcal{Y}} P(y \mid \theta^j, \mathbf{x}, \mathcal{D}_{t-1}) \log P(y \mid \theta^j, \mathbf{x}, \mathcal{D}_{t-1}) dy \tag{5.3}$$

and

$$\begin{aligned} H(y\ ;\ \mathbf{x}, \mathcal{D}_{t-1}) &= \int P(y \mid \mathbf{x}, \mathcal{D}_{t-1}) \log P(y \mid \mathbf{x}, \mathcal{D}_{t-1}) dy \qquad (5.4) \\ &\approx \int_{\mathcal{Y}} \left(\frac{1}{M} \sum_{j=1}^{M} P(y \mid \theta^j, \mathbf{x}, \mathcal{D}_{t-1}) \right) \\ &\quad \times \log \left(\frac{1}{M} \sum_{j=1}^{M} P(y \mid \theta^j, \mathbf{x}, \mathcal{D}_{t-1}) \right) dy. \qquad (5.5) \end{aligned}$$

Substituting equations 5.3 and 5.5 into the expression for mutual information (see equation 4.2), we obtain a convenient form for the mutual information that we use in our experiments:

$$I(\theta\ ;\ y \mid \mathbf{x}, \mathcal{D}_{t-1}) \approx \frac{1}{M} \sum_{j=1}^{M} D_{KL} \left(P(y \mid \theta^j, \mathbf{x}, \mathcal{D}_{t-1}) \,||\, P(y \mid \mathbf{x}, \mathcal{D}_{t-1}) \right), \tag{5.6}$$

where D_{KL} is the Kullback-Leibler (KL) divergence, a measure of how different one probability distribution is from another. Here,

$$\begin{aligned} &D_{KL} \left(P(y \mid \theta^j, \mathbf{x}, \mathcal{D}_{t-1}) \,||\, P(y \mid \mathbf{x}, \mathcal{D}_{t-1}) \right) \\ &= \int_{\mathcal{Y}} P(y \mid \theta^j, \mathbf{x}, \mathcal{D}_{t-1}) \log \frac{P(y \mid \theta^j, \mathbf{x}, \mathcal{D}_{t-1})}{P(y \mid \mathbf{x}, \mathcal{D}_{t-1})}. \end{aligned} \tag{5.7}$$

In our experiments for which $y \in \mathbb{R}$, we discretize y, allowing us to replace the integrals over y in the above expressions with sums.

Equation 5.6 makes clear that information-based active learning can be equivalently seen as comparing the prediction of the models given by each of the M samples, $P(y \mid \theta^j, \mathbf{x}, \mathcal{D}_{t-1})$, with the average model prediction $P(y \mid \mathbf{x}, \mathcal{D}_{t-1})$, and choosing the input that maximizes the average difference between predictions of individual models and the consensus model. This shows that infomax learning can also be seen as a form of query-by-committee (Settles, 2009). This sample-based formulation of infomax learning has also been referred to as Bayesian active learning by disagreement (BALD) (Houlsby et al., 2011; Gal et al., 2017).

5.2 Variational Inference (VI) Method. While Gibbs sampling allows us to accurately draw samples of the model parameter θ from its posterior $P(\theta \mid \mathcal{D}_{t-1})$, it is often slow and computationally inefficient. As an alternative, we therefore explored the use of variational inference (VI) (Blei et al., 2017) to compute a computationally efficient approximation to the posterior distribution over the model parameters $P(\theta \mid \mathcal{D}_{t-1})$. VI is typically faster than Gibbs sampling, but may be less accurate as it requires use of a simplified approximation to the posterior distribution over model parameters.

Here we use mean-field variational inference, which assumes that the model parameters and the latents are independent of each other,

$$q(\theta, z_{1:t-1}) = q_1(\theta) q_2(z_{1:t-1}), \tag{5.8}$$

where q_1 and q_2 represent the approximate variational posteriors over θ and $z_{1:t-1}$, respectively. We first assume simple tractable distributions to be the prior distributions over θ (such as a multivariate gaussian) and over the latents (such as an independent categorical distribution for z at every trial). We then use coordinate ascent to optimize the parameters of these assumed distributions in order to minimize the Kullback-Leibler divergence between the approximate and true posteriors:

$$q_1^*(\theta) q_2^*(z_{1:t-1}) = \underset{q_1^*(\theta) q_2^*(z_{1:t-1})}{\operatorname{argmin}} D_{KL}\left(q_1^*(\theta) q_2^*(z_{1:t-1}) \,||\, P(\theta, z_{1:t-1} \mid \mathcal{D}_{t-1})\right). \tag{5.9}$$

We describe the coordinate ascent update steps in detail for the model classes that we consider in the appendix (see sections A.2 and A.6).

However, having an approximate posterior over θ is insufficient to compute mutual information in closed form in the setting of discrete LVMs. The conditional response distribution $p(y_t \mid \mathbf{x}, \mathcal{D}_{t-1})$, which is required to compute the conditional entropy of y given θ, as well as the marginal entropy of y, is still not available in closed form. Hence, we instead draw samples of the model parameter $\{\theta^j\}_{j=1}^M$ from the variational posterior $q_1^*(\theta)$ (i.e., as

opposed to the true posterior, which we use in case of Gibbs sampling) and then use these samples to compute mutual information as described in equation 5.6. Overall, this approach is much faster than the MCMC sampling–based method because it does not require Gibbs sampling to obtain samples from the posterior over model parameters θ.

6 Mixture of Linear Regressions (MLR)

We now illustrate the power of our proposed infomax learning methods with applications to specific latent variable models, the first of which is a mixture of linear regressions (MLR) model. This simple model has a surprisingly rich history in machine learning (Li & Liang, 2018; Gaffney & Smyth, 1999; Bengio & Frasconi, 1995). It consists of an independent mixture of K distinct linear-gaussian regression models (see Figure 2A). Given an input, $\mathbf{x} \in \mathbb{R}^D$, the corresponding output observation $y \in \mathbb{R}$ arises from one of the K components as determined by the latent state $z \in \{1, ..K\}$. Formally, the model can be described as

$$z_t \sim \text{Cat}(\pi), \tag{6.1}$$

$$y_t \mid (\mathbf{x}_t, z_t = k) \sim \mathcal{N}(\mathbf{x}_t{}^\top \mathbf{w}_k, \sigma^2), \tag{6.2}$$

where $\pi \in \Delta^{K-1}$ denotes a discrete or categorical distribution over the set of K mixing components, $\mathbf{w}_k \in \mathbb{R}^D$ denotes the weights of the linear regression model in state k, and σ^2 denotes the observation noise variance. For simplicity, we assume here that σ^2 is known, so the model parameters to be learned are given by $\theta = \{\mathbf{w}_{1:K}, \pi\}$.

6.1 Fisher Information Analysis. Before applying our algorithm to the MLR model, it is worth asking whether there is any hope that infomax learning will be helpful in this setting. In the standard linear-gaussian regression model with gaussian prior, it is straightforward to show that posterior covariance of the model weights $\mathbf{w}$ is given by $(C_0^{-1} + \frac{1}{\sigma^2}\sum_{t=1}^{T} \mathbf{x}_t \mathbf{x}_t{}^\top)^{-1}$ where C_0 is the prior covariance. This expression is independent of the outputs $\{y_t\}$, which means that our uncertainty about the model parameters does not depend on the data we observe during the experiment. This implies that information gain is also independent of $\{y_t\}$, which means that an optimal design can be planned out prior to the experiment and there is no benefit to taking into account the output y_t on each trial when selecting the next input $\mathbf{x}_{t+1}$ (Chaloner, 1984; MacKay, 1992). Adaptive experimental design thus provides no benefit for the standard linear-gaussian regression model.

Intriguingly, however, we show that this does *not* hold for the MLR model; adaptive design can give large improvements over fixed designs for a *mixture* of linear models! We can analyze the potential benefits of

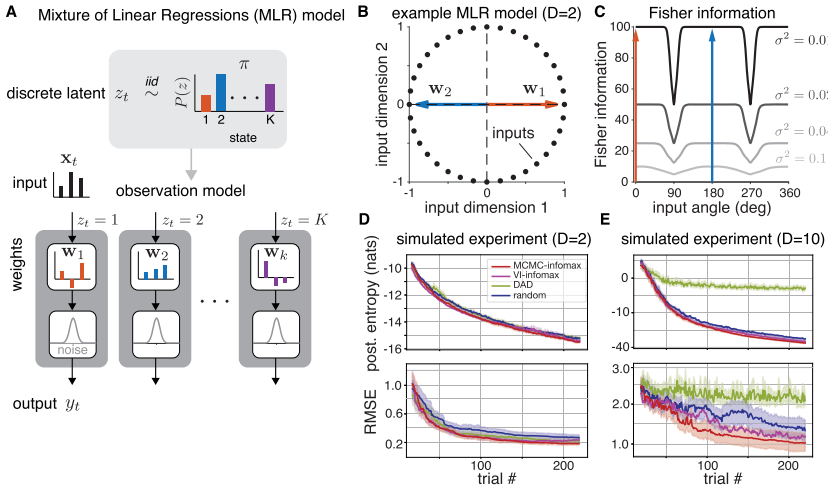

Figure 2. Infomax learning for mixture of linear regressions (MLR) models. (A) Model schematic. At time step t, the system is in state $z_t = k$ with probability π_k. The system generates output y_t using state-dependent weights $\mathbf{w}_k$ and independent additive gaussian noise (see equation 6.2). (B) Example two-state model with two-dimensional weights $\mathbf{w}_1 = (1, 0)$ and $\mathbf{w}_2 = (-1, 0)$. We consider possible inputs on the unit circle, which are the information-maximizing inputs for a linear gaussian model under an L_2 norm constraint. (C) Fisher information as a function of the angle between $\mathbf{w}_1$ and the input presented to the system for different noise variances σ^2. (D) Comparison between infomax active learning (using MCMC sampling and VI methods), DAD and random sampling for the 2D MLR model shown above with mixing probabilities $\pi = [0.6, 0.4]$ and noise variance $\sigma^2 = 0.1$. Error bars reflect 95% confidence interval (standard error) of the mean across 20 experiments. (E) Performance comparison for the same two-state model but with 10-dimensional weight vectors and inputs. The possible inputs to the system were uniform samples from the 10D unit hypersphere.

active learning by considering the Fisher information, which quantifies the asymptotic performance of an infomax learning algorithm (Paninski, 2005).

The Fisher information matrix for a model with parameters θ is a matrix with i, j'th element $J_{ij} = \mathbb{E}\left[\left(\frac{\partial}{\partial\theta_i}\log P(y \mid \mathbf{x}, \theta)\right)\left(\frac{\partial}{\partial\theta_j}\log P(y \mid \mathbf{x}, \theta)\right)\right]$, where expectation is taken with respect to $P(y \mid \mathbf{x}, \theta)$. For an MLR model in D dimensions with K components, the Fisher information matrix for the weights given an input vector $\mathbf{x}$ is a $KD \times KD$ matrix whose i, j'th block is given by

$$J_{[i,j]}(\mathbf{x}) = \frac{1}{\sigma^4}\mathbb{E}\big[(y - \mathbf{x}^\top\mathbf{w}_i)(y - \mathbf{x}^\top\mathbf{w}_j)P(z = i \mid y, \mathbf{x}, \theta)P(z = j \mid y, \mathbf{x}, \theta)\big]\,\mathbf{x}\mathbf{x}^\top, \tag{6.3}$$

where expectation is taken with respect to $P(y \mid \mathbf{x}, \theta)$, the marginal response distribution conditioned on the stimulus (see section A.4 for details). Although this expectation cannot generally be computed analytically (Behboodian, 1972), we can compute it for two extremal cases of interest: (1) perfect identifiability, when the response y gives perfect information about the latent variable z, and (2) perfect nonidentifiability, when the response provides no information about the latent variable.

To illustrate these two cases, Figure 2B shows an example MLR model with two 2D weight vectors pointing in opposite directions along the x_1-axis. If observation noise variance σ^2 is small, an input input $\mathbf{x} = [1, 0]$, corresponding to a unit vector with a 0 degree orientation, yields a response that makes the latent state perfectly identifiable, since the response will be large and positive if $z = 1$ and large and negative if $z = 2$. On the other hand, an input at 90 or 270 degrees gives rise to perfect nonidentifiability; these inputs are orthogonal to both $\mathbf{w}_1$ and $\mathbf{w}_2$, so observing the output y will provide no information about which model component (weights $\mathbf{w}_1$ or $\mathbf{w}_2$) produced it.

In the case of perfect identifiability, the Fisher information matrix simplifies to a block diagonal matrix with $\frac{1}{\sigma^2}\pi_i \mathbf{x}\mathbf{x}^\top$ in its ith diagonal block (see section A.4). The trace of the Fisher information matrix, which quantifies the total Fisher information provided by this input, is $\frac{1}{\sigma^2}||\mathbf{x}||^2$, which—remarkably—is the same Fisher information as in the standard (nonmixture) linear regression model. In the case of nonidentifiability, the Fisher information is a rank 1 matrix with block i, j given by $\frac{1}{\sigma^2}\pi_i\pi_j \mathbf{x}\mathbf{x}^\top$. In the case where all class prior probabilities are equal ($\pi_i = 1/K \ \forall i$), the trace is only $\frac{1}{K\sigma^2}||\mathbf{x}||^2$, revealing that nonidentifiable inputs can provide as little as $1/K$ as much Fisher information as inputs with perfect identifiability. The dependence on the number of components, K, is worth noting as it suggests that active learning yields larger improvements for models with more components.

Figure 2C shows the (numerically computed) Fisher information as a function of input angle for the MLR model shown in panel B, for different noise levels σ^2. This confirms the analytic result that Fisher information for this two-state MLR model is half its maximal value for inputs in the non identifiable region, and shows that this suboptimal region grows wider as noise variance increases. This analysis confirms that active learning can improve MLR model fitting and shows that the most informative inputs are those that provide information about the discrete latent variable.

6.2 Infomax Learning Algorithm for MLR. To quantify the potential usefulness of active learning for MLR models, we implemented both the MCMC-sampling and VI-based methods for infomax learning of the MLR model parameters $\theta = \{\mathbf{w}_{1:K}, \pi\}$. For the MCMC-sampling–based method, we used Gibbs sampling (Bishop, 2006) to obtain samples from the posterior over model parameters For the VI method, we updated the variational

parameters of the approximate posterior distribution after each trial, then drew samples of the model parameters from the variational posterior. (See sections A.1 and A.2 for details.) Thus, for both methods, we began by generating $M = 500$ samples of the model parameters, $\{\mathbf{w}_{1:K}^j, \pi^j\}_{j=1}^M$.

Then, to evaluate mutual information and select a stimulus, we computed the mutual information between the output y and the parameters θ for a grid of candidate inputs by substituting the likelihood term, $P(y \mid \theta^j, \mathbf{x}, \mathcal{D}_t) = \sum_{k=1}^K \pi_k^j \mathcal{N}(y \mid \mathbf{w}_k^j \cdot \mathbf{x}, \sigma^2)$, into equation 5.6. Finally, we selected the input $\mathbf{x}$ that maximized equation 5.6 and presented it to the system on the next trial.

6.3 Numerical Experiments for MLRs. We performed two different numerical experiments to evaluate our active learning framework for MLRs. In the first experiment, illustrated in Figure 2B, we considered a grid of possible inputs on the unit circle, spaced 10° apart. (This was motivated by the fact that the optimal stimuli for the linear regression model have maximal L_2 norm and thus lie on the surface of a hypersphere centered at zero.) On every trial, we selected an input from this set and sampled the output from one of $K = 2$ regression models. We fixed the state probabilities as $\pi = [0.6, 0.4]$. The regression models had the form $y_t = \mathbf{w}_k{}^\top \mathbf{x}_t + \epsilon$, where we fixed the generative parameters as $\mathbf{w}_1 = [-1, 0]$, $\mathbf{w}_2 = [1, 0]$, and $\epsilon \sim \mathcal{N}(0, 0.1)$.

Our second experiment followed the same setup, but we selected inputs from a set of 1000 candidate points sampled uniformly on the 10D hypersphere. The output again arose from one of the two regression models, now with weights oriented along the first two major axes, $\mathbf{w}_1 = [1, 0, \ldots, 0]$ and $\mathbf{w}_2 = [0, 1, 0 \ldots 0]$, again with mixing weights $\pi = [0.6, 0.4]$.

The task at hand is to learn the generative parameters of the model: $\{\mathbf{w}_1, \mathbf{w}_2, \pi\}$. We compared several input-selection strategies including our two (MCMC and VI-based) infomax learning methods, and the deep adaptive design (DAD) method proposed by Foster et al. (2021), as well as a random sampling method that selected inputs uniformly from the set of all possible inputs. For the DAD method, we adapted the publicly available code to use it for input selection in MLRs (details in section A.3). In all cases, after each trial, we used MCMC sampling–based inference to quantify performance of the various methods.

A natural quantity to track during infomax learning is the entropy of the posterior distribution over the model parameters θ (Bak & Pillow, 2018), which we approximate as $\log(|\mathrm{cov}(\theta)|)$ (we drop the additional term $\frac{D}{2}(1 + 2\pi)$ as it is constant for our experiments). We computed the sample estimate of this posterior entropy using the $M = 500$ samples obtained from MCMC sampling after every trial. We found that posterior entropy decreased fastest for our infomax algorithm with MCMC sampling (MCMC-infomax, top panel of Figure 2D). In 10D, this difference was even more

prominent (top panel of Figure 2E). We also tracked the root mean squared error (RMSE) between the true and estimated parameters. The bottom panel of Figure 2D shows that for the 2D simulation, RMSE decreased fastest for MCMC-infomax stimulus selection.

Finally, Figure 2E shows that in a model with 10D inputs, RMSE decreased fastest for MCMC-infomax, followed by infomax with variational inference (VI-infomax). This shows that evaluating information gain using samples from the true posterior produced substantially better learning than with samples from the variational posterior. Furthermore, while DAD was comparable to VI-infomax when learning two dimensions, it did not perform well for high-dimensional inputs. While DAD is a powerful general-purpose active learning method, these results emphasize the need for an active learning method tailored for discrete latent variable models. We feel these results were particularly impressive given that RMSE was *not* the objective function we optimized, as infomax-learning is distinct from learning algorithms with an MSE loss function. Overall, our proposed MCMC-infomax algorithm produced highly sample-efficient learning of MLRs in comparison to other methods.

In the case of 2D inputs, this improvement can be attributed to the fact that Fisher information drops dramatically when the angle between the weight vectors and the input is close to 90° or 270° (as discussed above). Hence, our infomax learning outperformed random sampling by avoiding the uninformative inputs orthogonal to the model weights. Figure 3 shows that our method did indeed avoid these inputs. As the Fisher information analysis given above makes clear, higher dimensionality leads to increased probability that randomly selected inputs will fall in the region of nonidentifiability (i.e., be orthogonal to all of the model weight vectors $\mathbf{w}_k$), given that random vectors in high dimensions have high probability of being orthogonal (Gorban & Tyukin, 2018). This aligns with our finding that the benefits of active learning are more pronounced in higher dimensions.

6.4 Application: California Housing Data Set. To examine performance in a real-world setting, we applied infomax learning to the California housing data set of Kelley Pace and Barry (1997). This data set contains median 1990 house prices for 20,640 census block groups along with eight predictors, and is accessible via scikit-learn (Pedregosa et al., 2011). We fit MLRs with different numbers of states to a reduced data set of 5000 samples and found that a three-state MLR described the California housing data set well (see Figure 4C) and offered a dramatic improvement in predictive power relative to standard linear regression (a one-state MLR). Figures 4A and 4B show the best-fitting mixing weights and state weights for this three state MLR.

Next, we wanted to understand if infomax learning would allow us to learn the best-fitting three state MLR parameters with fewer samples. Figures 4D and 4E show that MCMC-infomax learning did indeed

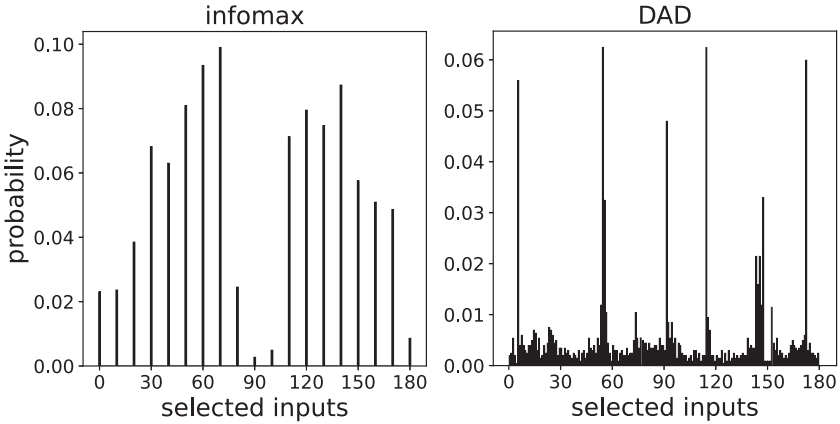

Figure 3. Left: Histogram showing the inputs selected by our MCMC-infomax active learning method for 200 trials using a mixture of linear regressions (MLR) model with inputs on a 2D circle (mutual information for MLRs is symmetric along the vertical axis in Figure 2; hence, we show only inputs in the range of 0 to 180°) This shows a drop in probability at 90°, which is predicted by our analysis of Fisher information (see Figure 2C). Right: Equivalent histogram for the DAD method, which did not show the same tendency to avoid inputs at 90°. Instead, the inputs selected covered the unit circle with modes appearing at multiples of approximately 30°. We are unsure why this is the case. (Note that DAD requires a continuous range of inputs; hence, it selected inputs on the entire unit circle as opposed to a discrete set.)

substantially reduce the number of samples required to learn the model parameters.

In Figure 4B, it is clear that the three discrete states differed most according to the weights placed on the AveOccup (average occupancy), Latitude, and Longitude covariates. Intriguingly, in Figure 4F, we see that the inputs selected by infomax learning had greater variance for the Latitude and Longitude covariates compared to those selected with random sampling (the red crosses are always above the blue dots). This is a useful external validation that infomax selects inputs in a manner that accords with intuition.

7 Hidden Markov Models

Hidden Markov models (HMMs) represent a class of structured discrete latent variable models that are richer and more powerful than the simple mixture models we have already considered. In a mixture model, the latent state is independent and identically distributed on each trial, meaning that the posterior distribution results from a product of conditionally independent likelihood terms. Reordering the data points would have no effect on

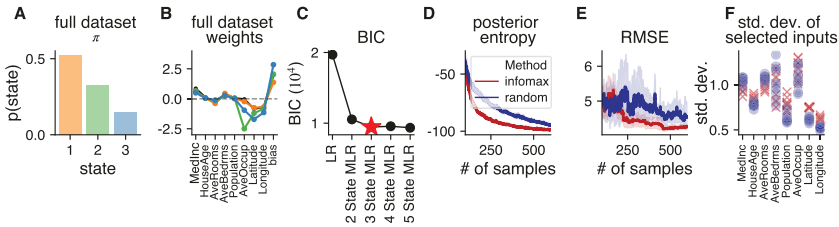

Figure 4. Application of infomax learning to California housing data set (Kelley Pace & Barry, 1997). (A) Best-fitting mixing weights for three-state MLR to 5000 samples of the data set. (B) Best-fitting state weights for three state MLR to 5000 samples of the California housing data set. Orange, green, and blue represent states 1, 2, and 3, respectively. Black represents the linear regression fit. (C) BIC as number of MLR states is varied from 1 (standard linear regression) to 5. We select the three-state model as BIC begins to level off beyond three states. (D) Posterior entropy between the three state MLR parameters obtained using 5000 samples (parameters shown in panels A and B) and recovered parameters as a function of the number of samples for random sampling (blue) and MCMC-infomax sampling (red). Error bars reflect 95% confidence interval of the mean across 10 experiments. (E) The same as in panel D but for the RMSE (root mean squared error). (F) Visualization of standard deviation of 500 inputs selected by both infomax (red) and random sampling (blue). Each dot corresponds to a different experiment. An examination of panel B makes it clear that the three states differ most according to the weights placed on the AveOccup, Latitude, and Longitude covariates. All 10 infomax experiments select inputs with greater variance for the latitude and longitude covariates than are selected by the random sampling experiments.

the posterior distribution over the model parameters. In an HMM, by contrast, the latent state depends on the state in the previous trial, which introduces sequential dependencies between the observations. HMMs therefore provide a natural modeling framework for systems whose states evolve in time (e.g., neurons, ecosystems, artificial and biological organisms). Learning the parameters of an HMM, however, requires large amounts of data, which makes them a natural candidate for active learning.

Here we consider HMMs designed for input-output data, in which the mapping between inputs and outputs depends on a latent state governed by an HMM. This general model family is commonly known as the input-output hidden Markov model (IO-HMM) (Bengio & Frasconi, 1995). In an IO-HMM with K discrete states, the observed output y_t depends on the current state, $z_t \in \{1, ..K\}$, as well as an input vector $\mathbf{x}_t \in \mathbb{R}^D$.

Recent work in neuroscience has focused on a class of IO-HMMs in which the input-output mapping is parameterized by a generalized linear model (GLM), resulting in a model known as the GLM-HMM (Escola et al.,

2011; Calhoun et al., 2019; Ashwood et al., 2022; Bolkan et al., 2022). Here, we consider the Bernoulli GLM-HMM, which assumes that the outputs are binary, $y_t \in \{0, 1\}$, and are produced according to state-specific GLM weights, $\mathbf{w}_k \in \mathbb{R}^D$:

$$P(y_t = 1 \mid \mathbf{x}_t, z_t = k) = \frac{1}{1 + \exp^{-\mathbf{w}_k{}^\top \mathbf{x}_t}}. \tag{7.1}$$

We assume that as in the standard HMM, state transitions are governed by a stationary, input-independent transition matrix, $A \in \mathbb{R}^{K \times K}$, where

$$A_{il} = P(z_t = l \mid z_{t-1} = i) \tag{7.2}$$

is the probability of transitioning from state i to state l on any trial. The first state z_1 has prior distribution $\pi \in \Delta^{K-1}$. The GLM-HMM model parameters are thus $\theta = \{\mathbf{w}_{1:K}, A, \pi\}$.

To perform infomax learning for GLM-HMMs, after each trial, we use Gibbs sampling to first iteratively sample the latent states $\{z_1, \ldots, z_t\}$ for all trials observed so far given the model parameters θ, and then sample the model parameters θ conditioned on these sampled latents (step 2 in Figure 1). Gibbs sampling-based inference for HMMs is well-known (Ghahramani, 2001). However, when we use Bernoulli-GLM observations, the conditional distribution over $\{\mathbf{w}_{1:K}\}$ is no longer available in closed form since there is no conjugate prior distribution for the weights of a Bernoulli GLM. Thus, we developed a method for sampling $\{\mathbf{w}_{1:K}\}$ using Laplace approximation (see section A.5 for details). An alternative strategy for sampling from logistic models involves using Polya-Gamma augmentation (Polson et al., 2013; Pillow & Scott, 2012). We compared these two approaches and found that our Laplace-based approach performed equally well to Polya-Gamma augmentation (see section A.7), thus empirically validating our method.

For comparison, we also developed an approximate infomax learning algorithm using variational inference (VI). We used mean-field VI to obtain posterior distributions over the model parameters θ. Because there is no conjugate prior for the GLM weights $\{\mathbf{w}_{1:K}\}$, we used the Laplace approximation to approximate their posteriors (see section A.6). After updating the variational posterior distribution on each time step, we drew samples of model parameters from their variational posteriors in order to evaluate the information gain associated with each candidate stimulus.

During infomax learning with GLM-HMMs (step 3 of Figure 1), we used $M = 500$ samples of the model parameters, $\{\mathbf{w}^j_{1:K}, A^j, \pi^j\}_{j=1}^M$, to compute the mutual information between the output and the model parameters according to equation 5.6. Here, the likelihood for the GLM-HMM is

$$P(y \mid \theta^j, x, \mathcal{D}_t) = \sum_{k=1}^{K} P(z = k \mid \mathcal{D}_t, \theta^j) P(y \mid x, z = k), \tag{7.3}$$

where $P(z = k \mid \mathcal{D}_t, \theta^j)$ can readily be obtained using the forward-backward algorithm and $P(y \mid x, z = k)$ is the Bernoulli-GLM likelihood function (see equation 7.1). We computed the mutual information over a discrete set of candidate inputs and then selected the most informative input to present on the subsequent trial.

7.1 Numerical Experiments with GLM-HMM. To test our active learning methods, we sampled data from the three-state GLM-HMM, shown in Figure 5B. We set the model parameters to closely approximate those from mice performing a binary sensory decision-making task in previous work (Ashwood et al., 2022). Each GLM has a weight (w_k) associated with the external stimulus as well as a bias parameter (b_k), such that the GLM weight vector is $\mathbf{w}_k = \{w_k, b_k\}$, and the input stimuli (x_t) are one-dimensional. In our experiment, we selected inputs from a grid of stimuli over the range $[-5, 5]$, spaced 0.01 units apart. Similar to the MLR setting, the task here is to recover the true parameters of the GLM-HMM used to simulate data. We compared the performance of our infomax learning methods (based on either MCMC sampling or variational inference) as well as a random sampling approach in which inputs were sampled uniformly at random. Deep adaptive design (DAD; Foster et al., 2021) is not applicable in this setting as it assumes trials to be independent and identically distributed (i.i.d.) and thus we did not consider it.

We examined the performance of these three methods and found that the posterior entropy over the model parameters decreased fastest under MCMC-infomax, followed by VI-infomax and was slowest with random sampling (see Figure 5C, left). We also observed that the RMSE between the true and inferred parameters decreased much faster for our active learning methods (with best performance under MCMC-infomax) as compared to random sampling for both the transition matrix A (middle panel of Figure 5C) and the GLM weights (right panel of Figure 5C). This suggests that our infomax learning method can be used to fit GLM-HMMs using fewer samples. It also reinforces our previous result that sampling from the exact posterior substantially benefits infomax learning as compared to using the variational posterior.

To understand why our framework outperforms random sampling for the GLM-HMM, we plotted histograms of the inputs selected by random sampling and by MCMC-infomax (see Figure 5D). While random sampling selected inputs from the entire input domain, infomax learning rarely selected inputs with a magnitude greater than 3. For more than three positive inputs, the sigmoid nonlinearity (see equation 7.1) is saturated for all three models, so that sampled y_t are 1 with high probability and are thus

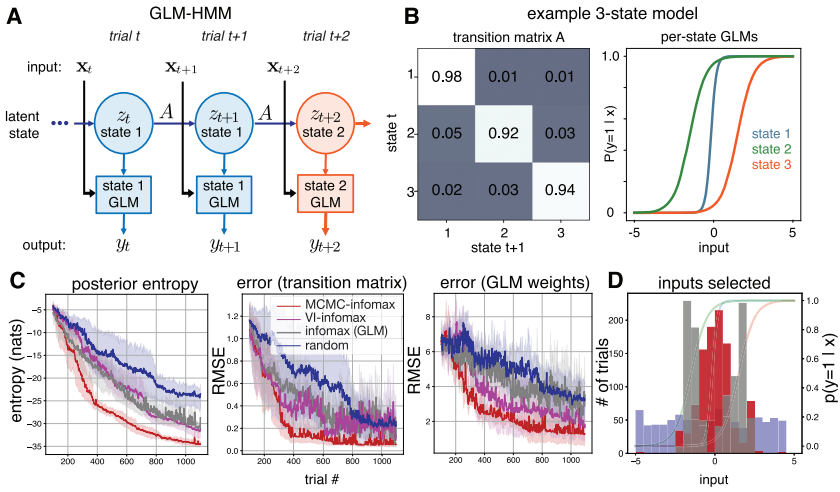

Figure 5. Infomax for GLM-HMMs. (A) Data generation process for the GLM-HMM. At time step t, the system generates output y_t based on its input $\mathbf{x}_t$ and the latent state z_t. The system then either remains in the same state or transitions into a new state at trial $t+1$, with the probabilities given by the entries in the transition matrix A. (B) Example settings for the transition matrix and state GLMs for a three-state GLM-HMM. These are the settings we use to generate output data for the analyses shown in panels C and D. (C) Left: Posterior entropy over the course of 1000 trials for random sampling (blue), infomax with a single GLM (gray), infomax for the full GLM-HMM using variational inference (VI) and MCMC sampling (magenta and red, respectively). Middle: Root mean squared error for the recovered transition matrix for each of the three input-selection schemes (random/infomax with GLM/infomax with GLM-HMM (MCMC)/infomax with GLM-HMM (VI)). Right: Root mean squared error for the weight vectors of the GLM-HMM for each of the input-selection schemes. (D) Selected inputs for random sampling (blue), active learning when there is model mismatch and the model used for infomax is a single GLM (gray), active learning with infomax (using MCMC sampling) and the full GLM-HMM (red). Selected inputs over the course of 1000 trials are plotted and are shown on top of the generative GLM curves.

uninformative about the latent state. Similarly, for large-magnitude negative inputs, the y_t samples are 0 with high probability for all three states. As such, the outputs generated by these provide virtually no information about the latents (necessary for updating the transition matrix) or the GLM weights. Overall, infomax learning substantially reduced the number of samples required to learn the parameters of the GLM-HMM.

To make our method practical for closed-loop experiments, it is critical for it to compute new inputs quickly. For example, in the case of mouse

decision-making experiments, consecutive trials occur within 1 to 10 seconds (Pinto et al., 2018; International Brain Laboratory et al., 2020). Our current implementation of infomax learning with Gibbs sampling (with a single chain of 500 samples) requires up to 6 seconds per trial (on an 8-core M2 chip laptop, or equivalently on a 32-core Intel Skylake node), while that of infomax learning with variational inference requires 3 to 4 seconds per trial. However, we show in section A.7 that running infomax with five parallel Gibbs chains of 100 samples each performs similar to a single 500-sample chain, and provides a 5 times speed-up requiring 1 to 2 seconds per trial. Additionally, we justify our choice of the length of the chain during Gibbs sampling (500 samples) in section A.7. These results provide further evidence that our infomax learning method is applicable across model settings.

7.2 Consequences of Ignoring Latent States. To assess the importance of latent structure on active learning methods, we benchmarked our method against an additional input-selection scheme: infomax under conditions of model mismatch. Specifically, we compared it to a strategy where inputs were selected by infomax under the (mismatched) assumption that responses arose from a single Bernoulli-GLM, with no latent states. This allowed us to explore the effect of ignoring the presence of latent variables when selecting inputs.

Figure 5D shows that the inputs selected by Bernoulli-GLM infomax learning differed substantially from those selected by the full GLM-HMM infomax algorithm. In particular, the Bernoulli-GLM method avoided selecting inputs in both the center and the outer edges of the input domain. By virtue of neglecting the outer edges, it outperformed random input selection (compare the gray and blue lines in all panels of Figure 5C). However, the full GLM-HMM infomax method still performed best for learning the weights and transition matrix of the true model (red lines in Figure 5C). The significant drop in the performance when ignoring the presence of latent states thus highlights the importance of developing active learning methods tailored specifically for latent variable models.

7.3 Downstream Application: Latent State Inference. GLM-HMMs are often used to infer the underlying latent states during the course of an experiment. To demonstrate the utility of our active learning approach for downstream tasks, we compared infomax learning and random sampling for predicting latent states across trials. We used the same generative GLM-HMM as shown in Figure 5B, and trained two new distinct GLM-HMMs using 400 input-output samples from the generative model. We trained one GLM-HMM using inputs selected by infomax learning (MCMC-based method) and another GLM-HMM using randomly selected inputs. Next, we generated a set of 100 trials from the generative model and used the two trained models to compute the posterior probability over the latents given the observed data. Figure 6 shows that the GLM-HMM trained using

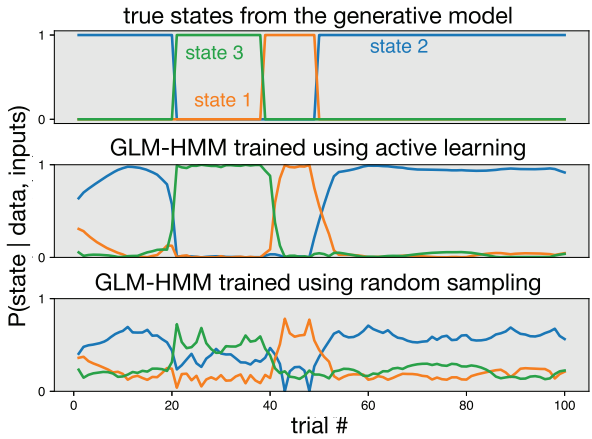

Figure 6. Inferring latent states. (Top) The true latent states of the data-generating GLM-HMM for 100 trials. (Middle) The posterior probabilities of states using a GLM-HMM trained using infomax learning on 400 trials from the data-generating GLM-HMM. (Bottom) The same for a GLM-HMM trained using random sampling on 400 trials from the data-generating GLM-HMM.

infomax learning was able to infer the true states far better than the model trained using random selection using the same number of trials.

7.4 Special Case: Mixture of GLMs. Finally, we evaluated infomax learning for a special case of GLM-HMMs: a simple mixture of Bernoulli-GLMs (MGLMs). Compared to standard GLM-HMMs, this model class assumes that the probability that the system transitions to state k at trial $t+1$ is independent of the system's state at trial t. MGLMs arise in a number of settings, including in medicine, transport modeling, and marketing (Farewell & Sprott, 1988; Follmann & Lambert, 1989, 1991; Wedel & DeSarbo, 1995; Li, 2018). Formally, MGLMs contain K distinct GLM observation models where the state of the model, $z \in \{1, ...K\}$, is independently sampled at each time step from a distribution $\pi \in \Delta^{K-1}$. Similar to the GLM-HMM setup, observations are generated according to a Bernoulli GLM as in equation 7.1. Infomax learning using Gibbs sampling for MGLMs involves steps similar to those required for GLM-HMMs and is described in section A.8.

We performed an experiment to assess the effectiveness of infomax learning in this setting. Data were generated from a two-state MGLM model (shown in Figure 7A) with $\pi = [0.6, 0.4]$ and the GLM weights $w_1 = [3, -6]$, $w_2 = [3, 6]$. We found that once again, our method outperformed random sampling-based learning and recovered the model parameters more accurately with fewer data points (see Figures 7B and 7C).

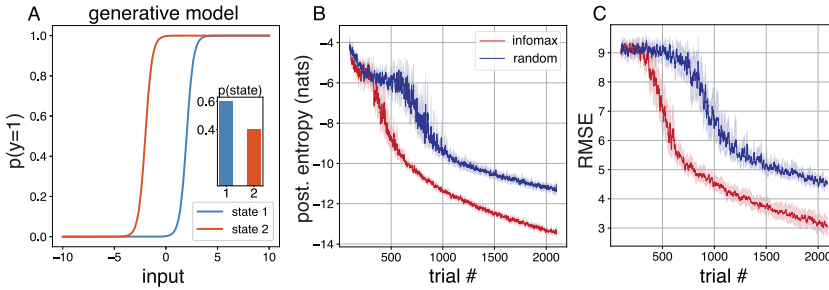

Figure 7. Infomax learning for mixture of GLMs (MGLMs). (A) Data generation model. Example settings for a two-state MGLM along with the mixing weights for the two states. (B) Posterior entropy of model parameters over the course of 2000 trials for random sampling (blue) and infomax learning (MCMC-sampling based) for MGLM (blue). (C) Root mean squared error for the recovered GLM weights and mixing weights for each of the two input-selection schemes.

8 Discussion

We have developed novel methods for Bayesian active learning in discrete latent variable models (LVMs). We applied these methods to two classes of models: mixture of linear regressions and input-output HMMs. We showed that infomax learning consistently achieved lower error and lower posterior entropy than random input selection. Our method also outperformed active learning methods that ignored the presence of latent variables and, for the case of MLRs, the DAD method of Foster et al. (2021).

MLRs represent a powerful class of models despite having a relatively simple mathematical formulation. This simplicity allowed us to theoretically validate our active learning method using Fisher information analysis and, furthermore, interpret the inputs selected by our method. While MLRs have not to our knowledge been previously applied in neuroscience, we feel they may have useful applications to data with real-valued observations, such as calcium imaging, EEG, MEG, fMRI, or behavioral pose modeling. For example, we expect our method to be applicable to modeling calcium imaging data in settings where those responses reflect a mixture of underlying causes or sources.

The success of our method in reducing the number of trials needed to fit a GLM-HMM suggests that it could be used to adaptively select stimuli during animal decision-making tasks. In traditional decision-making experiments, the experimenter selects a stimulus independently at random on each trial and then records the animal's decision in response to that stimulus. Fitting multistate GLM-HMMs (Ashwood et al., 2022; Bolkan et al., 2022) requires multiple sessions and days to accurately capture decision-making behavior. Using our framework, it may be possible to learn these

parameters using data from a single session. Our method can allow experimenters to adaptively select the most informative stimulus at every trial. This could reduce the time and cost of experiments and thereby speed up scientific discovery. Given the importance of LVMs in neuroscience (Escola et al., 2011; Calhoun et al., 2019; Ashwood et al., 2022; Bolkan et al., 2022; Yin et al., 2023) and other scientific domains, we envisage broad applicability of our method.

Finally, we briefly discuss several limitations of our work. First, we have only considered scalar as opposed to vector outputs. Extending to higher-dimensional outputs may require alternate methods for computing information, since the numerical integrals required for computing information-theoretic quantities are computationally intractable in high dimensions. Second, we selected maximally informative inputs from a discrete set of candidate inputs on each trial. Future work may instead use optimization to find optimal inputs in a continuous input space. A final direction for future work is to consider GLM-HMMs in which state transitions also depend on the input. Despite these limitations, our method substantially speeds up the learning of systems characterized by latent variable models and will be highly beneficial in neuroscience and other fields with time-consuming or expensive experiments.

Appendix

A.1 Gibbs Sampling for MLRs. Here we describe the Gibbs sampling algorithm for mixture of linear regressions models. Given T trials, for each input-output pair, $\mathbf{x}_t \in \mathbb{R}^D$ and $y_t \in \mathbb{R}$, we sample class belongings, $z_t \in \{1, ..K\}$, from

$$P(z_t = k \mid y_t, \mathbf{x}_t, \mathbf{w}_{1:K}, \pi, \sigma) = \frac{\mathcal{N}(y_t; \mathbf{w}_k{}^\top \mathbf{x}_t, \sigma^2)\pi_k}{\sum_l \mathcal{N}(y_t; \mathbf{w}_l{}^\top \mathbf{x}_t, \sigma^2)\pi_l}. \tag{A.1}$$

Next, we sample new estimates of the mixing parameters from

$$\pi_k \mid z_{1:T} \sim \mathrm{Dir}(n_k + 1), \tag{A.2}$$

where $n_k = \sum_{t=1}^{T} \mathbb{1}(z_{t'} = k)$.

Finally, we assume a gaussian prior, $\mathcal{N}(\mathbf{w}_0, \sigma_0^2 I)$, over the weights associated with each latent class and sample a new estimate for them as follows:

$$\mathbf{w}_k \sim \mathcal{N}(\mathbf{w}_k', \Sigma_k'), \tag{A.3}$$

$$\mathbf{w}_k' = \mathbf{w}_0 + (\sigma_0^2 I + X_k X_k{}^\top)^{-1} X_k{}^\top (Y_k - X_k \mathbf{w}_0), \tag{A.4}$$

$$\Sigma_k' = I - X_k{}^\top \left(\sigma_0^2 I + X_k X_k{}^\top\right)^{-1} X_k. \tag{A.5}$$

Here, the rows of $X_k \in T_k \times D$ and $Y_k \in T_k \times 1$ contain inputs and outputs at time points where $z = k$, respectively. We fix $\mathbf{w}_0 = \mathbf{0}$ and $\sigma_0^2 = 10$ in our experiments. We perform this procedure M times in order to obtain M samples of the model parameters, $\{\mathbf{w}_{1:K}^j, \pi^j\}_{j=1}^M$, where $M = 500$ (excluding 100 burn-in samples) in our experiments.

A.2 Variational Inference for MLRs. Here, we describe mean-field variational inference for MLRs, which we use to derive posterior distributions over the model's parameters. Following mean-field approximation, we assume independence between all the model parameters and the latent variables.

Given T trials, for each input-output pair, $\mathbf{x}_t \in \mathbb{R}^D$ and $y_t \in \mathbb{R}$, we assume that its mixture assignment $z_t \in \{1, ...K\}$ is governed by an independent categorical distribution $q(z_t; \phi_t)$ where $\phi_t \in \Delta^{K-1}$. We further assume that the weight $\mathbf{w}_k \in \mathbb{R}^D$ of the kth linear regression model has a normal posterior distribution $q(\mathbf{w}_k; \mu_k, \Sigma_k)$, with mean $\mu_k \in \mathbb{R}^D$ and covariance $\Sigma_k \in \mathbb{R}^{D\times D}$. Hence:

$$q(\mathbf{w}_{1:K}, z_{1:T}) = \prod_{t=1}^{T} q(z_t; \phi_t) \prod_{k=1}^{K} q(\mathbf{w}_k; \mu_k, \Sigma_k). \tag{A.6}$$

Let us vertically stack ϕ_t for $t \in 1 : T$ and denote this by a matrix ϕ of size $T \times K$. Similarly, let $X \in \mathbb{R}^{T\times D}$ represent the design matrix with all inputs stacked and $Y \in \mathbb{R}^{T\times 1}$ contain all observations. Also, we know that each of the linear regressions in the MLR model has gaussian noise with variance σ^2.

We update the variational parameters ϕ_t, $\mu_{1:K}$, and $\Sigma_{1:K}$ iteratively using the update rules described below. For each $t \in \{1..T\}$,

$$\phi_{tk} \propto \exp\{y_t {x_t}^\top \mathbb{E}[\mu_k] - \mathbb{E}[({x_t}^\top \mu_k)^2]/2\}. \tag{A.7}$$

Next, for each $k \in \{1...K\}$, we assume a gaussian prior distribution over the weights: $\mathcal{N}(\mathbf{w}_0, \sigma_0^2 I)$, we update the variational parameters governing the weights as follows:

$$\Sigma_k = \left(\sigma_0^2 I + \frac{1}{\sigma^2}\left((\phi_{:,k}1)\cdot X\right)^\top X\right)^{-1}, \tag{A.8}$$

$$\mu_k = \frac{1}{\sigma^2}\Sigma_k X^\top(\phi_{:,k}Y). \tag{A.9}$$

We fix $\mathbf{w}_0 = \mathbf{0}$ and $\sigma^2 = 10$ in our experiments. We repeat these updates until either the log-likelihood of the data arising from the model has converged or a limit of 500 iterations has been reached.

Once the variational posteriors have been learned, we draw M samples each for the weights $\mathbf{w}_{1:K}$ and the mixture assignments $z_{1:T}$. Finally, using the mixture assignments, we obtain M samples for the mixing probability π by computing the proportion of trials assigned to each state. We set $M = 500$ in our experiments, thus obtaining $\{\mathbf{w}_{1:K}^j, \pi^j\}_{j=1}^{500}$.

A.3 Training Details for Deep Adaptive Design (DAD). We downloaded the code for DAD and adapted it to perform input selection for MLRs. The parameters of the MLR model were set to the same values as described in section 6.3. Since the DAD model requires continuous inputs rather than a discrete list of inputs, we allow it to choose inputs from the unit circle in 2D and the unit hypersphere in 10D, rather than restricting it to the discrete set of stimuli in section 6.3.

The DAD model has two components: the encoder network, which takes in input-observation pairs $\{x, y\}$ and outputs an encoding for this. This is a feedforward neural network. We set this network to have three layers: the input layer, which has three nodes for the first MLR experiment (2D inputs and 1D observations) and 11 nodes for the second experiment (10D inputs and 1D observations), a hidden layer with 256 nodes and ReLU activation function, and a linear output layer with 16 nodes.

Following this, the encoded history is taken as input by an emitter network. This network outputs the input for the next trial, x_t. The input layer of this feedforward network has the same dimensionality as the output of the embedding layer: 16 nodes. It has one hidden layer with ReLU activation and 256 nodes, followed by a linear output layer with as many nodes as the dimensionality of the input to the MLR model. We normalize the output of this network to ensure that the selected x_t lies on the unit circle/unit hypersphere.

We do a hyperparameter optimization to select the number of hidden layers and nodes from the range of values used in the experiments (number of hidden layers: 1–3; number of nodes per layer: 16/128/256) in the original DAD (Foster et al., 2021) paper.

To compute the sPCE loss that DAD uses to optimize the two neural networks, we use 500 samples each to compute the inner and outer expectations in the loss function. Since our experiments involve large number of trials ($T = 200$), we use a score gradient estimator to compute the gradients that are backpropagated while training. Finally, we train the model using Adam (with betas set to 0.8, 0.998) and use exponential learning rate annealing (where the initial learning rate is set to 1e-4 post a search over the range 1e-5–1e-3, and $\gamma = 0.96$) for a total of 50,000 gradient steps.

A.4 Fisher Information for MLRs. Here we derive the Fisher information for the weights of the MLR model (shown in Figure 2B).

We consider a model consisting of a mixture of K linear regression models in a D-dimensional input space, defined by weights $\{\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_K\}$. The full model weights take the form of a length-KD vector formed by stacking the weights for each component:

$$\mathbf{w} = \begin{bmatrix} \mathbf{w}_1 \\ \vdots \\ \mathbf{w}_K \end{bmatrix}. \tag{A.10}$$

The Fisher information J is a $KD \times KD$ matrix carrying the expectation for the product of partial derivatives of the log-likelihood with respect to each element of $\mathbf{w}$. We will derive the $D \times D$ blocks of the Fisher information matrix for each pair of components in $\{1, \ldots, K\}$.

The block of partial derivatives for component j is given by

$$\begin{aligned} \frac{\partial}{\partial \mathbf{w}_j} \log p(y \mid \mathbf{x}, \theta) &= \frac{1}{\sigma^2}(y - \mathbf{x}^\top \mathbf{w}_j)\mathbf{x} \left(\frac{\pi_j \exp\left(-\frac{1}{2\sigma^2}(y - \mathbf{x}^\top \mathbf{w}_j)^2\right)}{P(y \mid \mathbf{x}, \theta)} \right) \\ &= \frac{1}{\sigma^2}(y - \mathbf{x}^\top \mathbf{w}_j)\mathbf{x} \left(\frac{P(y \mid \mathbf{x}, z = j, \theta) P(z = j \mid \pi)}{P(y \mid \mathbf{x}, \theta)} \right) \\ &= \frac{1}{\sigma^2}(y - \mathbf{x}^\top \mathbf{w}_j)\mathbf{x} \Big(P(z = j \mid y, \mathbf{x}, \theta) \Big). \end{aligned} \tag{A.11}$$

Plugging this into the formula for Fisher information, we obtain the following expression for the i, jth block of the Fisher information matrix:

$$J_{[i,j]}(\mathbf{x}) = \frac{1}{\sigma^4} \mathbb{E}\big[(y - \mathbf{x}^\top \mathbf{w}_i)(y - \mathbf{x}^\top \mathbf{w}_j) P(z = i \mid y, \mathbf{x}, \theta) P(z = j \mid y, \mathbf{x}, \theta)\big] \mathbf{x}\mathbf{x}^\top, \tag{A.12}$$

where expectation is taken with respect to the marginal distribution $P(y \mid \mathbf{x}, \theta)$. This expectation cannot in general be computed in closed form (see Behboodian, 1972). However, we considered two special cases in the text where an analytic expression is available.

A.4.1 Perfect Identifiability. First, the case of perfect identifiabilty arises when the conditional distributions $P(y \mid \mathbf{x}, z = j, \theta)$ are well separated for the different classes of latent variable z or, equivalently, the posterior class probabilities $P(z = j \mid y, \mathbf{x}, \theta)$ are effectively 0 or 1 for virtually all output values y. In practice, this arises for inputs $\mathbf{x}$ such that the conditional means $\{\mathbf{x}^\top \mathbf{w}_1, \mathbf{x}^\top \mathbf{w}_2, \ldots, \mathbf{x}^\top \mathbf{w}_K\}$ are well separated relative to the noise standard deviation σ (e.g., more than 2σ apart). In this case, the off-diagonal blocks of the Fisher information matrix are zero, since $P(z = i \mid y, \mathbf{x}, \theta) P(z = j \mid y, \mathbf{x}, \theta) \approx 0$ for $i \neq j$. The diagonal blocks, by contrast, can be computed in

closed form:

$$\begin{aligned} J_{[j,j]}(\mathbf{x}) &= \frac{1}{\sigma^4}\mathbb{E}\big[(y - \mathbf{x}^\top \mathbf{w}_j)^2 P(z = j \mid y, \mathbf{x}, \theta)^2\big]\mathbf{x}\mathbf{x}^\top \\ &= \frac{1}{\sigma^4}\left(\int_{-\infty}^{\infty}(y - \mathbf{x}^\top \mathbf{w}_j)^2 \pi_j \mathcal{N}(y \mid \mathbf{x}^\top w, \sigma^2) dy\right)\mathbf{x}\mathbf{x}^\top \\ &= \frac{1}{\sigma^2}\pi_j \mathbf{x}\mathbf{x}^\top. \end{aligned} \tag{A.13}$$

We can write the Fisher information matrix efficiently as

$$J(\mathbf{x}) = \frac{1}{\sigma^2}\mathrm{diag}(\pi) \otimes \mathbf{x}\mathbf{x}^\top, \tag{A.14}$$

where $\otimes$ denotes the Kronecker product. The trace of the Fisher information is

$$\mathrm{Tr}[J] = \frac{1}{\sigma^2}\mathrm{Tr}[\mathrm{diag}(\pi)]\mathrm{Tr}[\mathbf{x}\mathbf{x}^\top] = \frac{1}{\sigma^2}\mathbf{x}^\top \mathbf{x}, \tag{A.15}$$

which is the trace of the Fisher information matrix in the standard linear-gaussian regression model. This confirms—as one might expect—that in the case of perfect identifiability, we have the same amount of Fisher information as in a model without latent variables.

A.4.2 Nonidentifiability. Second, the case of non identifiability arises when the conditional distributions $P(y \mid \mathbf{x}, z = j, \theta)$ are identical for the different classes of latent variable z, meaning the output y carries no information about the mixing component that generated it. This arises when the linear projection of $\mathbf{x}$ onto all of the weight vectors is identical, $\mathbf{x}^\top \mathbf{w}_1 = \mathbf{x}^\top \mathbf{w}_2 = \cdots = \mathbf{x}^\top \mathbf{w}_K$. This arises, for example, when the stimulus is orthogonal to all of the weight vectors, which occurs with high probability in high-dimensional settings.

In this case, we can also compute the Fisher information in closed form. We obtain, for block i, j of the Fisher information matrix,

$$\begin{aligned} J_{[i,j]}(\mathbf{x}) &= \frac{1}{\sigma^4}\mathbb{E}\big[(y - \mathbf{x}^\top \mathbf{w}_i)^2 \pi_i \pi_j\big]\mathbf{x}\mathbf{x}^\top \\ &= \frac{1}{\sigma^2}\pi_i \pi_j \mathbf{x}\mathbf{x}^\top, \end{aligned} \tag{A.16}$$

where we have used the fact that $\mathbf{x}^\top \mathbf{w}_i = \mathbf{x}^\top \mathbf{w}_j$ and that the product of posterior probabilities $P(z = i \mid y, \mathbf{x}, \theta)P(z = j \mid y, \mathbf{x}, \theta)$ is equal to the product of prior probabilities $\pi_i \pi_j$ in the setting where the output y carries no information about the latent z.

Algorithm 1: GLM-HMM Gibbs Sampling.

Input: Observations $y_{1:T}$, Inputs $\mathbf{x}_{1:T}$, Prior hyperparameters: α, $\mathbf{w}_0$, σ_0
Output: Samples $\{(z_{1:T}, \mathbf{w}_{1:K}, A, \pi)^{(j)}\}$

Initialize $z_{1:T}$, $\mathbf{w}_{1:K}$, A, π
for $j \leftarrow 1, ...M$ **do**
  for $k \leftarrow 1, ...K$ **do**
    $\mathbf{w}_k^j \leftarrow \text{GLMSAMPLEPOSTERIOR}(\{y_t, \mathbf{x}_t \mid z_t = k\}_{1:T}, \mathbf{w}_0, \sigma_0, \mathbf{w}_k^{j-1})$
    $A_{k,:}^j \leftarrow \text{sample Dir}(\alpha_{k,:} + \mathbf{n}_{k,:})$ $\quad \rightarrow$ where $n_{kl} = \sum_t \mathbf{I}(z_t = k, z_{t+1} = l)$
  $z_{1:T}^j \leftarrow \text{IOHMMSAMPLESTATE}(\pi, A, L)$ $\quad \rightarrow$ s.t. $L_{t,k} = P(y_t \mid \mathbf{x}_t, \mathbf{w}_k)$
  $\pi^j \leftarrow \text{sample Dir}(\alpha_{0,:} + \mathbb{I}_{z_1})$

The Fisher information matrix can be written in Kronecker form,

$$J = \frac{1}{\sigma^2}(\pi\pi^\top) \otimes \mathbf{x}\mathbf{x}^\top, \tag{A.17}$$

which has trace

$$\text{Tr}[J] = \frac{1}{\sigma^2}(\pi^\top\pi)\mathbf{x}^\top\mathbf{x}. \tag{A.18}$$

This expression is minimal when the prior probabilities are all equal to $1/K$, in which case $\pi^\top\pi = 1/K$, giving $\text{Tr}[J] = \frac{1}{K\sigma^2}\mathbf{x}^\top\mathbf{x}$.

A.5 Gibbs Sampling for GLM-HMMs. We provide a complete description of Gibbs sampling for GLM-HMMs in algorithm 1. It uses outputs $y_{1:T}$ and inputs $\mathbf{x}_{1:T}$, along with the prior over model parameters to provide M samples of the latent states $\{z_{1:T}\}^j$ as well as of the model parameters $\{\mathbf{w}_{1:K}, A, \pi\}^j$. We assume the model has K distinct latent states. Sampling the latent states (see algorithm 3) requires using backward messages, $B_{t,k} = P(y_{t+1:T} \mid x_{1:T}, z_t = k)$, which can be obtained using a standard forward-backward algorithm (Bishop, 2006). To sample the weights of the GLMs per state, we use the Laplace approximation followed by an acceptance-rejection step detailed in algorithm 2. We fix the Dirichlet prior $\alpha \in \mathbb{R}^{K+1\times K}$ over the rows of the transition matrix, A, and the initial state distribution, π, to be a matrix of ones. The GLM weights have an identical prior: $\mathcal{N}(\mathbf{0}, 10)$. Further, we run Gibbs sampling for 500 iterations and discard the first 100.

A.6 Variational Inference for GLM-HMMs. For a GLM-HMM with K distinct states and Bernoulli-GLM observations, we want to learn variational posteriors for the initial state distribution $\pi_0 \in \Delta^{K-1}$, the transition matrix $A \in \mathbb{R}^{K\times K}$, and the weights of the GLMs, $\mathbf{w}_{1:K} \in \mathbb{R}^D$. To do so, we

Algorithm 2 : GLM Sample Weight from Posterior.

Input: Observations $y_{1:T'}$, Inputs $x_{1:T'}$, Prior: $\mathbf{w}_0, \sigma_0$, Previous estimate of $\mathbf{w}$: $\mathbf{w}^{\text{old}}$
Output: $\{\mathbf{w}\}$
function GLMSAMPLEPOSTERIOR($(y_{1:T'}, x_{1:T'}, \mathbf{w}_0, \sigma_0, \mathbf{w}^{\text{old}})$)
  $L(\mathbf{w}) = \sum_{t=1}^{T'} \log P(y = y_t \mid x_t, \mathbf{w})$
  $\mathbf{w}^{\text{MAP}} \leftarrow \text{argmax}_{\mathbf{w}} \left(L(\mathbf{w}) + \log \mathcal{N}(\mathbf{w}; \mathbf{w}_0, \sigma_0^2 I)\right)$
  $C \leftarrow -\left(\frac{\partial^2 L(\mathbf{w})}{d\mathbf{w}^2} - \sigma_0^{-2} I\right)^{-1} \Big|_{\mathbf{w}_{\text{MAP}}}$
  $\mathbf{w}^* \leftarrow$ sample $\mathcal{N}(\mathbf{w}^{\text{MAP}}, C)$
  $\alpha(\mathbf{w}^*, \mathbf{w}^{\text{old}}) \leftarrow \min\left(1, \frac{\tilde{p}(\mathbf{w}^* | y_{1:T'}, x_{1:T'}) \mathcal{N}(\mathbf{w}^{old}; \mathbf{w}^{\text{MAP}}, C)}{\tilde{p}(\mathbf{w}^{old} | y_{1:T'}, x_{1:T'}) \mathcal{N}(\mathbf{w}^*; \mathbf{w}^{\text{MAP}}, C)}\right)$ $\rightarrow \tilde{p}$: unnormalized posterior
  if $\alpha(\mathbf{w}^*, \mathbf{w}^{\text{old}}) \geq U(0,1)$ **then**
    $\mathbf{w} \leftarrow \mathbf{w}^*$
  else
    $\mathbf{w} \leftarrow \mathbf{w}^{\text{old}}$

Algorithm 3: GLM-HMM State Sequence Sampling.

Input: Initial state dist. π, Transition matrix A, Likelihood matrix $L \in \mathbb{R}^{T \times K}$
Output: $z_{1:T}$
function IOHMMSAMPLESTATE((π, A, L))
  $B \leftarrow$ HMM-Backwardmessages(A, L) $\rightarrow B_{t,k} = P(y_{t+1:T} \mid x_{1:T}, z_t = k)$ (Bishop, 2006)
  $z_1 \leftarrow$ sample $\pi_k B_{1,k} L_{1,k}$ over $k \in \{1, \ldots K\}$

  for $t \leftarrow 2, \ldots T$ **do**
    $z_t \leftarrow$ sample $A_{z_{t-1},k} B_{t,k} L_{t,k}$ over $k \in \{1, \ldots K\}$

use inputs to the model $\mathbf{x}_{1:T}$ and their corresponding observations $y_{1:T}$. The unknown latent states corresponding to these trials are represented by $z_{1:T}$.

We first define prior distributions over the model parameters:

$$\pi_0 \sim Dir(\boldsymbol{\alpha}_0), \tag{A.19}$$

$$A_{j,:} = \pi_j \sim Dir(\boldsymbol{\alpha}_j) \quad j = 1 \ldots K, \tag{A.20}$$

$$\mathbf{w}_k \sim \mathcal{N}(\mathbf{w}_0, \sigma_0^2 I) \quad k = 1 \ldots K, \tag{A.21}$$

where $\boldsymbol{\alpha}_0 \in \mathbb{R}^K$ and $\boldsymbol{\alpha}_j \in \mathbb{R}^K$ and contain positive real numbers only, $\mathbf{w}_0 \in \mathbb{R}^D$, $\sigma_0 \in \mathbb{R}$. Now we define a variational posterior over the parameters and latent states of the GLM-HMM as follows:

$$q(z_{1:T}, A, \pi_0, \phi_{k=1}^K) = q(z_1) \prod_{t=2}^{T} q(z_t \mid z_{t-1}) q(A) q(\pi_0) \prod_{k=1}^{K} q(\phi_k). \tag{A.22}$$

Here, we assume that the latents are independent of the model parameters, which reflects the mean-field assumption. Next, we develop a coordinate ascent algorithm to iteratively learn the variational posteriors.

We initialize $q(\pi_0),\ q(A),\ q(\mathbf{w}_k)$ to their prior distributions. Then, in the first step, we compute the following quantities:

$$\tilde{\pi}_0 = \exp\{\mathbb{E}_{q(\pi_0)}[\ln \pi_0]\}, \tag{A.23}$$

$$\tilde{A}_{j,:} = \exp\{\mathbb{E}_{q(A)}[\ln A_{j,:}]\}, \tag{A.24}$$

$$\tilde{L}_{t,k} = \exp\{\mathbb{E}_{q(\mathbf{w}_k)}[\ln P(y_t \mid \mathbf{w}_k, \mathbf{x}_t)]\} = \exp\left\{\frac{1}{N}\sum_{i=1}^{N} \ln P(y_t \mid \mathbf{w}_k^i, \mathbf{x}_t)\right\}. \tag{A.25}$$

The Dirichlet distributions over π_0 and $A_{j,:}$ provide closed-form updates for $\tilde{\pi}_0$ and $\tilde{A}_{j,:}$ (in particular, for a D-dimensional vector $x \sim Dir(\boldsymbol{\gamma})$, $\mathbb{E}[\ln x_i] = \psi(\gamma_i) - \psi(\sum_i \gamma_i)$, where ψ is the digamma function). To compute $\tilde{L}_{t,k}$, which is not available in closed form in the case of GLM observations, we obtain a sample estimate of the expectations using 10 samples.

Next, using the quantities computed above, we run a forward-backward algorithm for GLM-HMMs (Bishop, 2006) and obtain the forward and backward messages $F,\ B \in \mathbb{R}^{T\times K}$. This leads to the following distributions over the latent states:

$$q(z_t = k) = F_{t,k}B_{t,k} / \left(\sum_{k'} B_{T,k'}\right), \tag{A.26}$$

$$q(z_{t-1} = j, z_t = k) = F_{t-1,j}\tilde{A}_{j,k}\tilde{L}_{t,k}B_{t,k} / \left(\sum_{k'} B_{T,k'}\right). \tag{A.27}$$

Now we are ready to update the variational distributions over the model parameters:

$$q(\pi_0) \propto \prod_{k=1}^{K} \pi_{0k}^{\alpha_{0k}+q(z_1=k)-1}, \tag{A.28}$$

$$q(A) \propto \prod_{k=1}^{K} \pi_{jk}^{\alpha_{jk}+\sum_{t=2}^{T} q(z_{t-1}=j, z_t=k)-1}. \tag{A.29}$$

And finally, the variational approximation over the GLM weights is

$$q(\mathbf{w}_k) \propto \exp\left\{\sum_{t=1}^{T} q(z_t = k)\ln P(y_t \mid \mathbf{w}_k, \mathbf{x}_t) + \ln P(\mathbf{w}_k)\right\}. \tag{A.30}$$

Unlike typical gaussian HMMs, this is not available in closed form because the likelihood of a Bernoulli-GLM does not have a conjugate prior. To deal with this, we approximate $q(\mathbf{w}_k)$ by a gaussian distribution using Laplace approximation. Let $L(\mathbf{w}_k) = \exp\left\{\sum_{t=1}^{T} q(z_t = k) \ln P(y_t \mid \mathbf{w}_k, \mathbf{x}_t) + \ln P(\mathbf{w}_k)\right\}$,

$$q(\mathbf{w}_k) \sim \mathcal{N}(\mathbf{w}'_k, \Sigma'_k); \quad \mathbf{w}'_k = \text{argmax}_{\mathbf{w}_k} L(\mathbf{w}_k), \quad \Sigma'_k = \left(\frac{\partial^2 L(\mathbf{w}_k)}{\partial \mathbf{w}_k^2}\right)^{-1}\Bigg|_{\mathbf{w}'_k}. \tag{A.31}$$

We repeat the update equations from equation A.23 to equation A.31 iteratively until the log-likelihood of the data from the model converges or a maximum of 500 iterations is reached.

Once we have obtained a variational distribution for all the model parameters, we can draw M samples of $\{\pi_0^j, A, \mathbf{w}_{1:K}^j\}_{j=1}^M$ from their variational posteriors. We set $M = 500$ for our experiments.

A.7 Additional Analyses for GLM-HMMs. Here, we compare our infomax learning method using variants of Gibbs sampling. In all our experiments in section 7.1, we run a single chain to obtain 500 samples of the model's parameters, discarding the initial 200 burn-in samples. If we instead run five parallel chains, each of length 140, and discard the first 40 samples as burn-in, we would still be able to obtain 500 samples of the model parameters to perform infomax learning, but this provides a 5 times improvement in speed, leading to 1 to 2 seconds per trial for input selection. We verify in Figure 8 that the perform of infomax while using parallel chains of Gibbs is comparable to that using a single long chain (compare the red and violet traces).

Finally, in all our experiments, we use our Laplace-based Gibbs sampling approach for GLM-HMMs (detailed in section A.5). We compared this to Polya-Gamma augmented Gibbs sampling (Polson et al., 2013; Pillow & Scott, 2012), an established technique in the literature to sample from logistic models. In this case, weights of the GLM are sampled using Polya-Gamma augmentation, while the strategy for sampling the latents and the state transitions remains the same as in algorithm 1. We show in Figure 8 that our approach is comparable to Polya-Gamma augmentation in terms of both posterior entropy and error in recovering the model parameters (compare the peach and red curves). This empirically verifies the utility of our Laplace-based Gibbs sampling approach for GLM-HMMs.

We also varied the length of the chain used during our Laplace-based Gibbs sampling approach to find the optimal number of samples needed for accurately fitting the model and selecting the next input at every trial. We varied the length of a single Gibbs chain between 125 and 1000 samples.

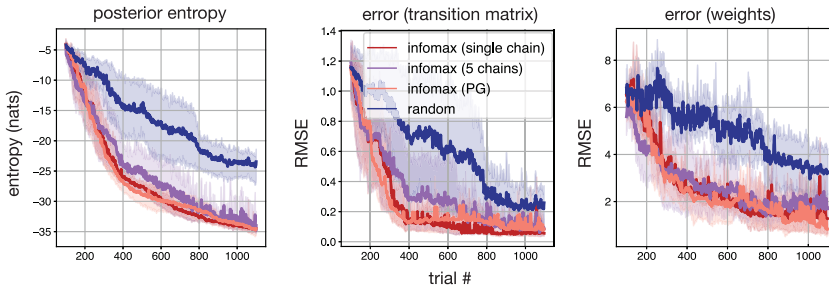

Figure 8. Infomax learning for GLM-HMMs. (Left) The posterior entropy of model parameters over the course of 1000 trials when performing infomax learning using our Laplace-based Gibbs sampling approach with a single long chain (red), using parallel chains of our Laplace-based Gibbs sampler (violet), using Polya-Gamma augmented Gibbs sampling (peach), and using random sampling (blue). (Middle, right) Shows error in recovering the transition matrix and the weights of the GLMs using the same set of methods.

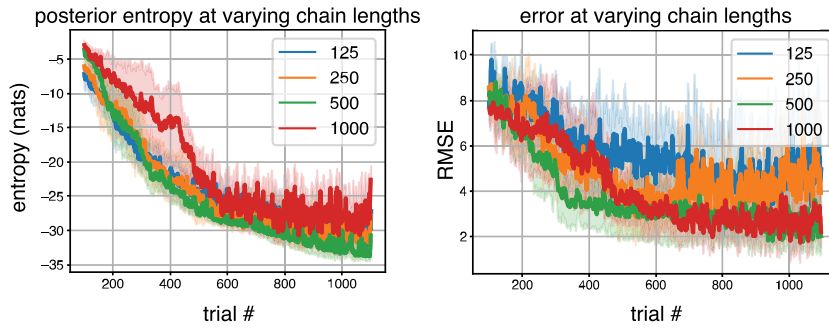

Figure 9. Gibbs sampling-based infomax learning for GLM-HMMs, with varying lengths of a single Gibbs chain. Left panel shows the posterior entropy of model parameters over the course of 1000 trial using Gibbs sampling with a single long chain. Each trace shows posterior entropy when using a different number of samples obtained from Gibbs sampling, the same samples are used to select the next input. Error bars correspond to 95% confidence interval of the mean over 5 experiments. Right panel shows error in recovering model parameters, while varying the number of samples in the Gibbs chain.

In each case, we discarded the first 100 samples as burn-in and used the rest to select the best input for the next trial. We find that chains of length 500 are optimal in terms of the root mean squared error between true and predicted model parameters, as well as posterior covariance (see Figure 9).

Algorithm 4: MGLMs Gibbs Sampling.

Input: Observations $y_{1:T}$, Inputs $\mathbf{x}_{1:T}$, Priors: α_0, $\mathbf{w}_0$, σ_0
Output: Samples $\{(z_{1:T}, \mathbf{w}_{1:K}, \pi)^{(j)}\}$
Initialize $z_{1:T}$, $\mathbf{w}_{1:K}$, A, π
for $j \leftarrow 1, ...M$ **do**
  for $k \leftarrow 1, ...K$ **do**
    $\mathbf{w}_k^j \leftarrow \text{GLMSAMPLEPOSTERIOR}(\{y_t, \mathbf{x}_t \mid z_t = k\}_{1:T}, \mathbf{w}_0, \sigma_0, \mathbf{w}_k^{j-1})$
  $\pi^j \leftarrow$ sample $\text{Dir}(\alpha_0 + \mathbf{n})$ $\quad \rightarrow$ where $n_k = \sum_t \mathbf{I}(z_t = k)$
  $z_t^j \leftarrow$ sample $P(z_t \mid y_t, \mathbf{x}_t) \forall\ t = \{1 : T\}$ $\quad \rightarrow$ s.t. $P(z_t = k \mid y_t, \mathbf{x}_t) = \frac{P(y=y_t|\mathbf{x}_t,\mathbf{w}_k)\pi_k}{\sum_k P(y=y_t|\mathbf{x}_t,\mathbf{w}_k)\pi_k}$

A.8 Gibbs Sampling for MGLMS. Gibbs sampling for MGLMs is similar to that for GLM-HMMs except that now the states can be sampled independently of each other. Algorithm 4 provides full details. We set a Dirichlet prior over the initial state distribution, with $\alpha_0 = \mathbf{1} \in \mathbb{R}^K$, and that over the weights to be $\mathcal{N}(\mathbf{0}, 10)$. Here, we run Gibbs sampling for 700 iterations and discard the first 200 as burn-in (MGLMs require a longer burn-in period).

Acknowledgments

This work was supported by a Google Ph.D. fellowship (to A.J.) and grants from the Simons Collaboration on the Global Brain (SCGB AWD543027), the NIH BRAIN initiative (NS104899 and 9R01DA056404-04), and a U19 NIH-NINDSBRAIN Initiative Award (5U19NS104648). We thank Benjamin Cowley and Orren Karniol-Tambour for useful comments and discussions at various points during this project.

References

Anderson, B., & Moore, A. (2005). Active learning for hidden Markov models: Objective functions and algorithms. In *Proceedings of the 22nd International Conference on Machine Learning* (pp. 9–16).

Ashwood, Z. C., Roy, N. A., Stone, I. R., Urai, A. E., Churchland, A. K., Pouget, A., & Pillow, J. W. (2022). Mice alternate between discrete strategies during perceptual decision-making. *Nature Neuroscience*, *25*(22), 201–212. 10.1038/s41593-021-01007-z

Bak, J. H., Choi, J., Witten, I., Akrami, A., & Pillow, J. W. (2016). Adaptive optimal training of animal behavior. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, & R. Garnett (Eds.), *Advances in neural information processing systems, 29* (pp. 1939–1947). Curran.

Bak, J. H., & Pillow, J. W. (2018). Adaptive stimulus selection for multialternative psychometric functions with lapses. *Journal of Vision*, *18*(12), 4. 10.1167/18.12.4

Behboodian, J. (1972). Information matrix for a mixture of two normal distributions. *Journal of Statistical Computation and Simulation1*(4), 295–314. 10.1080/00949657208810024

Bengio, Y., & Frasconi, P. (1995). An input output HMM architecture. In G. Tesauro, D. S. Touretzky, & T. K. Leen (Eds.), *Advances in neural information processing systems, 7* (pp. 427–434). MIT Press.

Bishop, C. M. (2006). *Pattern recognition and machine learning*. Springer.

Blei, D. M., Kucukelbir, A., & McAuliffe, J. D. (2017). Variational inference: A review for statisticians. *Journal of the American Statistical Association*, *112*(518), 859–877. 10.1080/01621459.2017.1285773

Bolkan, S. S., Stone, I. R., Pinto, L., Ashwood, Z. C., Iravedra Garcia, J. M., Herman, A. L., Singh, P., . . . Witten, I. B. (2022). Opponent control of behavior by dorsomedial striatal pathways depends on task demands and internal state. *Nature Neuroscience*, *25*(33), 345–357. 10.1038/s41593-022-01021-9

Calhoun, A. J., Pillow, J. W., & Murthy, M. (2019). Unsupervised identification of the internal states that shape natural behavior. *Nature Neuroscience*, *22*(12), 2040–2049. 10.1038/s41593-019-0533-x

Chaloner, K. (1984). Optimal Bayesian experimental design for linear models. *Annals of Statistics*, *12*(1), 283–300. 10.1214/aos/1176346407

Chaloner, K., & Verdinelli, I. (1995). Bayesian experimental design: A review. *Statistical Science*, *10*(3), 273–304. 10.1214/ss/1177009939

Chen, Z., Vijayan, S., Barbieri, R., Wilson, M. A., & Brown, E. N. (2009). Discrete- and continuous-time probabilistic models and algorithms for inferring neuronal up and down states. *Neural Computation*, *21*(7), 1797–1862. 10.1162/neco.2009.06-08-799

Cohn, D. A., Ghahramani, Z., & Jordan, M. I. (1996). Active learning with statistical models. *Journal of Artificial Intelligence Research*, *4*, 129–145. 10.1613/jair.295

Cover, T., & Thomas, J. (1991). *Elements of information theory*. Wiley.

Cowley, B., Williamson, R., Clemens, K., Smith, M., & Byron, M. Y. (2017). Adaptive stimulus selection for optimizing neural population responses. In I. Guyon, Y. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, & R. Garnett (Eds.), *Advances in neural information processing systems, 30* (pp. 1395–1405). Curran.

DiMattina, C. (2015). Fast adaptive estimation of multidimensional psychometric functions. *Journal of Vision*, *15*(9), art. 5.

DiMattina, C., & Zhang, K. (2011). Active data collection for efficient estimation and comparison of nonlinear neural models. *Neural Computation*, *23*(9), 2242–2288. 10.1162/NECO_a_00167

DiMattina, C., & Zhang, K. (2013). Adaptive stimulus optimization for sensory systems neuroscience. *Frontiers in Neural Circuits*, *7*. 10.3389/fncir.2013.00101

Escola, S., Fontanini, A., Katz, D., & Paninski, L. (2011). Hidden Markov models for the stimulus-response relationships of multistate neural systems. *Neural Computation*, *23*(5), 1071–1132. 10.1162/NECO_a_00118

Farewell, V. T., & Sprott, D. A. (1988). The use of a mixture model in the analysis of count data. *Biometrics*, *44*(4), 1191–1194. 10.2307/2531746

Follmann, D. A., & Lambert, D. (1989). Generalizing logistic regression by nonparametric mixing. *Journal of the American Statistical Association*, *84*(405), 295–300. 10.1080/01621459.1989.10478769

Follmann, D. A., & Lambert, D. (1991). Identifiability of finite mixtures of logistic regression models. *Journal of Statistical Planning and Inference*, *27*(3), 375–381. 10.1016/0378-3758(91)90050-O

Foster, A., Ivanova, D. R., Malik, I., & Rainforth, T. (2021). Deep adaptive design: Amortizing sequential Bayesian experimental design. In *Proceedings of the 38th International Conference on Machine Learning* (pp. 3384–3395).

Gaffney, S., & Smyth, P. (1999). Trajectory clustering with mixtures of regression models. In *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 63–72).

Gal, Y., Islam, R., & Ghahramani, Z. (2017). *Deep Bayesian active learning with image data*. arXiv:1703.02910.

Ghahramani, Z. (2001). An introduction to hidden Markov models and Bayesian networks. *International Journal of Pattern Recognition and Artificial Intelligence*, *15*, 9–42. 10.1142/S0218001401000836

Glaser, J. I., Whiteway, M. R., Cunningham, J. P., Paninski, L., & Linderman, S. W. (2020). *Recurrent switching dynamical systems models for multiple interacting neural populations*. bioRxiv.

Gollisch, T., & Herz, A. V. (2012). The iso-response method: Measuring neuronal stimulus integration with closed-loop experiments. *Frontiers in Neural Circuits*, *6*. 10.3389/fncir.2012.00104

Gorban, A. N., & Tyukin, I. Y. (2018). Blessing of dimensionality: Mathematical foundations of the statistical physics of data. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, *376*(2118), 20170237.

Hefang, L., Myers, R. H., & Keying, Y. (2000). Bayesian two-stage optimal design for mixture models. *Journal of Statistical Computation and Simulation*, *66*(3), 209–231. 10.1080/00949650008812023

Houlsby, N., Huszár, F., Ghahramani, Z., & Lengyel, M. (2011). *Bayesian active learning for classification and preference learning*. arXiv:1112.5745.

International Brain Laboratory, Aguillon-Rodriguez, V., Angelaki, D. E., Bayer, H. M., Bonacchi, N., Carandini, M., . . . Zador, A. (2020). *A standardized and reproducible method to measure decision-making in mice*. bioRxiv:2020.01.17.909838.

Ivanova, D. R., Foster, A., Kleinegesse, S., Gutmann, M. U., & Rainforth, T. (2021). *Implicit deep adaptive design: Policy-based experimental design without likelihoods*. arXiv:1112.5745:2111.02329.

Jha, A., Morais, M. J., & Pillow, J. W. (2021). Factor-analytic inverse regression for high-dimension, small-sample dimensionality reduction. In *Proceedings of the 38th International Conference on Machine Learning* (pp. 4850–4859).

Kelley Pace, R., & Barry, R. (1997). Sparse spatial autoregressions. *Statistics and Probability Letters*, *33*(3), 291–297. 10.1016/S0167-7152(96)00140-X

Kemere, C., Santhanam, G., Yu, B. M., Afshar, A., Ryu, S. I., Meng, T. H., & Shenoy, K. V. (2008). Detecting neural-state transitions using hidden Markov models for motor cortical prostheses. *Journal of Neurophysiology*, *100*(4), 2441–2452. 10.1152/jn.00924.2007

Khuri, A. I., Mukherjee, B., Sinha, B. K., & Ghosh, M. (2006). Design issues for generalized linear models: A review. *Statistical Science*, *21*(3), 376–399. 10.1214/088342306000000105

Kim, W., Pitt, M. A., Lu, Z.-L., Steyvers, M., & Myung, J. I. (2014). A hierarchical adaptive approach to optimal experimental design. *Neural Computation*, *26*(11), 2465–2492. 10.1162/NECO_a_00654

Kirsch, A., van Amersfoort, J., & Gal, Y. (2019). Batchbald: Efficient and diverse batch acquisition for deep Bayesian active learning. In H. Wallach, H. Larochelle, A. Beygelzimer, F. Alché-Buc, E. Fox, & R. Garnett (Eds.) *Advances in neural information processing systems*, *32*. Curran.

Kleinegesse, S., & Gutmann, M. U. (2020). Bayesian experimental design for implicit models by mutual information neural estimation. In *Proceedings of the 37th International Conference on Machine Learning* (pp. 5316–5326).

Kuck, H., de Freitas, N., & Doucet, A. (2006). SMC samplers for Bayesian optimal nonlinear design. In *Proceedings of the 2006 IEEE Nonlinear Statistical Signal Processing Workshop* (pp. 99–102).

Lewi, J., Butera, R., & Paninski, L. (2007). Efficient active learning with generalized linear models. In *Proceedings of the International Conference on Artificial Intelligence and Statistics* (pp. 267–274).

Lewi, J., Butera, R., & Paninski, L. (2009). Sequential optimal design of neurophysiology experiments. *Neural Computation*, *21*(3), 619–687. 10.1162/neco.2008.08-07-594

Lewi, J., Schneider, D. M., Woolley, S. M. N., & Paninski, L. (2011). Automating the design of informative sequences of sensory stimuli. *Journal of Computational Neuroscience*, *30*(1), 181–200. 10.1007/s10827-010-0248-1

Li, G. (2018). Application of finite mixture of logistic regression for heterogeneous merging behavior analysis. *Journal of Advanced Transportation*, *2018*, e1436521.

Li, Y., & Liang, Y. (2018). Learning mixtures of linear regressions with nearly optimal complexity. In *Proceedings of the 31st Conference on Learning Theory* (pp. 1125–1144).

Linderman, S. W., Johnson, M. J., Wilson, M. A., & Chen, Z. (2016). A Bayesian nonparametric approach for uncovering rat hippocampal population codes during spatial navigation. *Journal of Neuroscience Methods*, *263*, 36–47. 10.1016/j.jneumeth.2016.01.022

MacKay, D. J. C. (1992). Information-based objective functions for active data selection. *Neural Computation*, *4*(4), 590–604. 10.1162/neco.1992.4.4.590

Miller, P., & Katz, D. B. (2010). Stochastic transitions between neural states in taste processing and decision-making. *Journal of Neuroscience*, *30*(7), 2559–2570. 10.1523/JNEUROSCI.3047-09.2010

Myung, J. I., Cavagnaro, D. R., & Pitt, M. A. (2013). A tutorial on adaptive design optimization. *Journal of Mathematical Psychology*, *57*(3–4), 53–67. 10.1016/j.jmp.2013.05.005

Paninski, L. (2005). Asymptotic theory of information-theoretic experimental design. *Neural Computation*, *17*(7), 1480–1507. 10.1162/0899766053723032

Park, M., Weller, J. P., Horwitz, G. D., & Pillow, J. W. (2014). Bayesian active learning of neural firing rate maps with transformed gaussian process priors. *Neural Computation*, *26*(8), 1519–1541. 10.1162/NECO_a_00615

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., . . . Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Pillow, J., & Scott, J. (2012). Fully Bayesian inference for neural models with negative-binomial spiking. In F. Pereira, C. J. Burges, L. Bottou, & K. Q. Weinburger (Eds.), *Advances in neural information processing systems*, *25*. Curran.

Pillow, J. W., & Park, M. (2016). Adaptive Bayesian methods for closed-loop neurophysiology. In A. El Hady (Ed.), *Closed loop neuroscience* (pp. 3–18). Elsevier.

Pinto, L., Koay, S. A., Engelhard, B., Yoon, A. M., Deverett, B., Thiberge, S. Y., . . . Brody, C. D. (2018). An accumulation-ofevidence task using visual pulses for mice navigating in virtual reality. *Frontiers in Behavioral Neuroscience*, *12*. 10.3389/fnbeh.2018.00036

Polson, N. G., Scott, J. G., & Windle, J. (2013). Bayesian inference for logistic models using Polya–Gamma latent variables. *Journal of the American Statistical Association*, *108*(504), 1339–1349. 10.1080/01621459.2013.829001

Rainer, G., & Miller, E. K. (2000). Neural ensemble states in prefrontal cortex identified using a hidden Markov model with a modified EM algorithm. *Neurocomputing*, *32*, 961–966. 10.1016/S0925-2312(00)00266-6

Roy, N., & McCallum, A. (2001). Toward optimal active learning through Monte Carlo estimation of error reduction. In *Proceedings of the 25th International Conference on Machine Learning* (pp. 441–448).

Ryan, E. G., Drovandi, C. C., McGree, J. M., & Pettitt, A. N. (2016). A review of modern computational algorithms for Bayesian optimal design. *International Statistical Review/Revue Internationale de Statistique*, *84*(1), 128–154.

Seeger, M. W. (2008). Bayesian inference and optimal design for the sparse linear model. *Journal of Machine Learning Research*, *9*, 759–813.

Seeger, M. W., & Nickisch, H. (2008). Compressed sensing and Bayesian experimental design. In *Proceedings of the 25th International Conference on Machine Learning* (pp. 912–919).

Settles, B. (2009). *Active learning literature survey.* Computer Sciences Technical Report 1648, University of Wisconsin–Madison.

Shababo, B., Paige, B., Pakman, A., & Paninski, L. (2013). Bayesian inference and online experimental design for mapping neural microcircuits. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, & K. Q. Weinberger (Eds.), *Advances in neural information processing systems*, *26* (pp. 1304–1312). Curran.

Steinke, F., Seeger, M., & Tsuda, K. (2007). Experimental design for efficient identification of gene regulatory networks using sparse Bayesian models. *BMC Systems Biology*, *1*(1), 51. 10.1186/1752-0509-1-51

Vasisht, D., Damianou, A., Varma, M., & Kapoor, A. (2014). Active learning for sparse Bayesian multilabel classification. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 472–481).

Verdinelli, I., & Kadane, J. B. (1992). Bayesian designs for maximizing information and outcome. *Journal of the American Statistical Association*, *87*(418), 510–515. 10.1080/01621459.1992.10475233

Watson, A. B. (2017). Quest+: A general multidimensional Bayesian adaptive psychometric method. *Journal of Vision*, *17*(3), 10.

Watson, A., & Pelli, D. (1983). QUEST: A Bayesian adaptive psychophysical method. *Perception and Psychophysics*, *33*, 113–120. 10.3758/BF03202828

Wedel, M., & DeSarbo, W. S. (1995). A mixture likelihood approach for generalized linear models. *Journal of Classification*, *12*(1), 21–55. 10.1007/BF01202266

Weilnhammer, V., Stuke, H., Eckert, A.-L., Standvoss, K., & Sterzer, P. (2021). *Humans and mice fluctuate between external and internal modes of sensory processing*. bioRxiv.

Wiltschko, A. B., Johnson, M. J., Iurilli, G., Peterson, R. E., Katon, J. M., Pashkovski, S. L., . . . Datta, S. R. (2015). Mapping sub-second structure in mouse behavior. *Neuron*, *88*(6), 1121–1135. 10.1016/j.neuron.2015.11.031

Wu, D., Niu, R., Chinazzi, M., Vespignani, A., Ma, Y.-A., & Yu, R. (2021). *Deep Bayesian active learning for accelerating stochastic simulation*. arXiv:2106.02770.

Yin, C., Melin, M. D., Rojas-Bowe, G., Sun, X. R., Gluf, S., Couto, J., . . . Churchland, A. K. (2023). *Engaged decision-makers align spontaneous movements to stereotyped task demands.* bioRxiv:2023.06.26.546404.

Yu, B. M., Cunningham, J. P., Santhanam, G., Ryu, S. I., Shenoy, K. V., & Sahani, M. (2009). Gaussian-process factor analysis for low-dimensional single-trial analysis of neural population activity. *Journal of Neurophysiology*, *102*(1), 614.

Zoltowski, D. M., Pillow, J. W., & Linderman, S. W. (2020). A general recurrent state space framework for modeling neural dynamics during decision-making. In H. Daumé & A. Singh (Eds.), *Proceedings of the 37th International Conference on Machine Learning* (pp. 11680–11691).

Zucchini, W., Raubenheimer, D., & MacDonald, I. L. (2008). Modeling time series of animal behavior by means of a latent-state model with feedback. *Biometrics*, *64*(3), 807–815. 10.1111/j.1541-0420.2007.00939.x

Received June 2, 2023; accepted October 13, 2023.