SIMPL: SCALABLE AND HASSLE-FREE OPTIMIZATION OF NEURAL REPRESENTATIONS FROM BEHAVIOUR

Tom M George
Sainsbury Wellcome Centre, UCL
tom.george.20@ucl.ac.uk

Pierre Glaser
Gatsby Computational Neuroscience Unit, UCL

Kimberly Stachenfeld
Google DeepMind & Columbia University

Caswell Barry
Dept. of Cell and Developmental Biology, UCL

Claudia Clopath
Sainsbury Wellcome Centre, UCL & Imperial College London
c.clopath@imperial.ac.uk

ABSTRACT

High-dimensional neural activity in the brain is known to encode low-dimensional, time-evolving, behaviour-related variables. A fundamental goal of neural data analysis consists of identifying such variables and their mapping to neural activity. The canonical approach is to assume the latent variables *are* behaviour and visualize the subsequent tuning curves. However, significant mismatches between behaviour and the encoded variables may still exist — the agent may be thinking of another location, or be uncertain of its own — distorting the tuning curves and decreasing their interpretability. To address this issue a variety of methods have been proposed to learn this latent variable in an unsupervised manner; these techniques are typically expensive to train, come with many hyperparameters or scale poorly to large datasets complicating their adoption in practice. To solve these issues we propose SIMPL (Scalable Iterative Maximization of Population-coded Latents), an EM-style algorithm which iteratively optimizes latent variables and tuning curves. SIMPL is fast, scalable and exploits behaviour as an initial condition to further improve convergence and identifiability. We show SIMPL accurately recovers latent variables in biologically-inspired spatial and non-spatial tasks. When applied to a large rodent hippocampal dataset SIMPL efficiently finds a modified latent space with smaller, more numerous, and more uniformly-sized place fields than those based on behaviour, suggesting the brain may encode space with greater resolution than previously thought.

1 INTRODUCTION

Large neural populations in the brain are known to encode low-dimensional, time-evolving latent variables which are, oftentimes, closely related to behaviour (Afshar et al., 2011; Harvey et al., 2012; Mante et al., 2013; Carnevale et al., 2015; Kobak et al., 2016). Coupled with a recent data-revolution driven by the advent of large-scale neural recording techniques (Jun et al., 2017; Wilt et al., 2009), focus in recent years has shifted from single-cell to population-level analyses where the goal is to extract these variables using a variety of statistical (Yu et al., 2008a; Cunningham & Yu, 2014; Kobak et al., 2016; Zhao & Park, 2017; Williams et al., 2020) and computational (Van der Maaten & Hinton, 2008; Pandarinath et al., 2018; Mackevicius et al., 2019) methods, ultimately providing deeper insight into the computations embodied by neural circuits.

This paradigm shift is particularly pertinent in the context of the mammalian spatial memory system where Nobel-prize winning discoveries have identified cells whose neural activity depends on spatially-relevant behavioural variables such as position (O'Keefe & Dostrovsky, 1971; O'Keefe, 1978; Hafting et al., 2005; Doeller et al., 2010; Moser et al., 2015), heading direction (Taube et al., 1990), speed (McNaughton et al., 1983) and distance to environmental boundaries (Lever et al.,

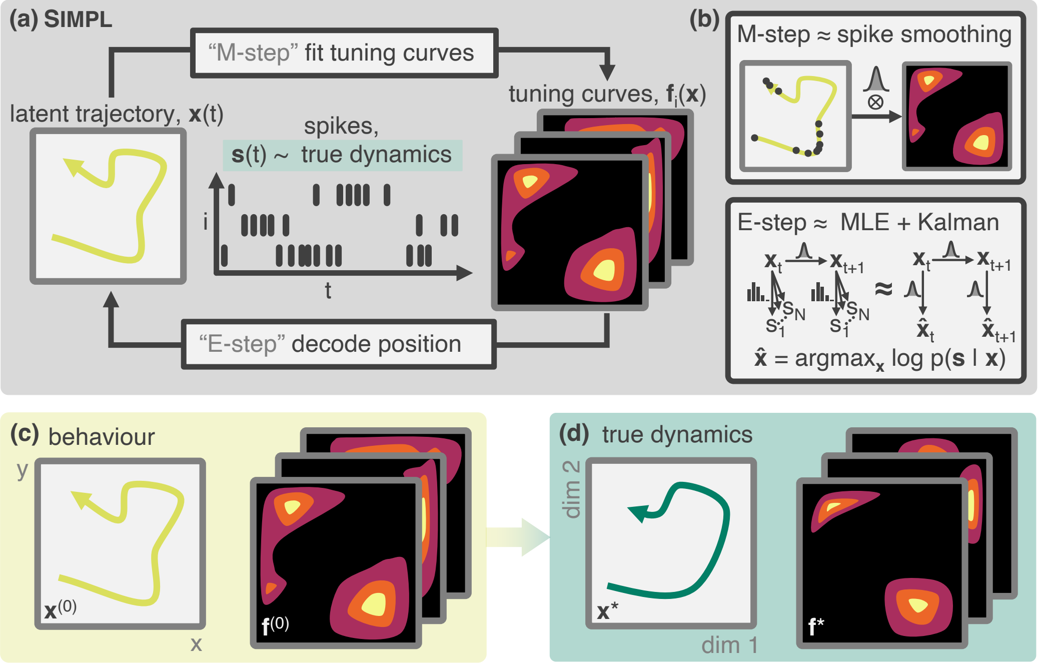

Figure 1: Schematic of the SIMPL algorithm. **(a)** A latent variable model for spiking data $(\mathbf{f}_i(\mathbf{x}), \mathbf{x}(t))$ is optimized by iterating a two-step procedure closely related to the expectation-maximization (EM, Dempster et al. 1977) algorithm: First, tuning curves are fitted to an initial estimate of the latent variable (the "M-step"), which are then used to *re*decode the latent variable (the "E-step"). **(b)** SIMPL fits tuning curves using kernel density estimation (KDE) with a Gaussian kernel (top) and decodes the latent variables by Kalman-smoothing maximum likelihood estimates. Measured behaviour **(c)** is used to initialize the algorithm as it is often closely related to the true generative latent variable of interest **(d)**.

2009)/objects (Høydal et al., 2019)in a highly structured manner. These discoveries include place cells (O'Keefe & Dostrovsky, 1971) and grid cells (Hafting et al., 2005) which are widely held to constitute the brain's "cognitive map" (Tolman, 1948; O'Keefe, 1978). Characterizing neural activity in terms of behaviour has been, and remains, a cornerstone practice in the field; however, the core assumption supporting it — that the latent variable encoded by neural activity *is* the behavioural variable — is increasingly being called into question (Sanders et al., 2015; Whittington et al., 2020; George et al., 2024b).

The brain is not a passive observer of the world. Active internal processing like planning a future route (Spiers & Maguire, 2006) or recalling past positions (Squire et al., 2010) as well as observed phenomena such as replay (Carr et al., 2011), theta sweeps (Maurer et al., 2006), and predictive coding (Muller & Kubie, 1989; Mehta et al., 1997; Stachenfeld et al., 2017) will cause encoded variables to deviate from behaviour. Additionally, the brain is not a perfect observer; irreducible uncertainty due to limited, noisy or ambiguous sensory data can lead to similar encoding discrepancies. Experimental inaccuracies, like measuring the wrong behaviour or measuring behaviour poorly, can contribute further. These hypotheses are supported by decoding analyses which show that "behaviour" decoded from behaviourally-fitted tuning curves rarely achieves perfect performance (Wilson & McNaughton, 1993) as well as the observation that neurons show high variability under identical behavioural conditions (Fenton & Muller, 1998; Low et al., 2018). All combined, these facts hint at a much richer and more complex internal code. When this complexity is not accounted for (as is typically the case), neural data may be misinterpreted and tuning curves will be blurred or distorted relative to their true form, weakening the validity of the conclusions drawn from them. Nonetheless, the observation that behaviour is still a close-but-imperfect proxy for the true latent variable motivates the search for techniques to *refine* behaviourally fitted tuning curves as opposed to starting from scratch. Current methods either fail to exploit behaviour (Yu et al., 2008a; Wu et al., 2017), don't scale to large neural data sets (Wu et al., 2017), are computationally expensive to train (Smith & Brown, 2003; Pandarinath et al., 2018) or are limited in the expressiveness of their tuning curve models (Macke et al., 2011; Gao et al., 2016; Archer et al., 2014).

Contributions Here we introduce SIMPL (Scalable Iterative Maximization of Population-coded Latents), a straightforward yet effective enhancement to the current paradigm. Our approach fits tuning curves to observed behaviour and iteratively refines these through a two-step process: first we *decode* the latent variable from the previously estimated tuning curves; then, we *refit* the curves based on these decoded latents. SIMPL imposes minimal constraints on the structure of the tuning curves, scales well to large neural datasets and does not rely on neural network function approximators which can be hard to interpret and expensive to train. We theoretically analyse SIMPL and establish formal connections to Expectation-Maximisation (EM, Dempster et al. 1977) for a simple but flexible class of generative models. By exploiting behaviour as an initialization, SIMPL converges fast and alleviates well known issues to do with local minima and identifiability (Hyvärinen & Pajunen, 1999; Locatello et al., 2019). This allows it to reliably return refined tuning curves and latent variables which remain close to, but improve upon, their behavioural analogues readily admitting direct comparison.

We first validate and analyse the properties of SIMPL on synthetic datasets that closely match those analysed by experimentalists: a discrete 2AFC decision-making task and a continuous grid cells dataset. Finally, we apply SIMPL to rodent electrophysiological hippocampal data Tanni et al. (2022) and show it modifies the latent space in an incremental but significant way. The optimized tuning curves explain the data better than their behavioural counterparts and contain sharper, more numerous place fields which allow for a reinterpretation of previous experimental results, motivating the use of SIMPL in future studies. SIMPL has only two hyperparameters and can be run on quickly on large neural datasets $\mathcal{O}$(1 hour, 200 neurons, 10^6 spikes)$\sim$ $\mathcal{O}$(1 min) without requiring a GPU. It outperforms a popular modern alternative technique based on neural networks (Schneider et al., 2023) and is over 30× faster. This make it a practical alternative to existing tools particularly of interest to navigational communities where data is abundant and behavioural variables are beneficial. We provide an open-source JAX-optimised (Bradbury et al., 2018) implementation of our code[1].

2 METHOD

Here we provide a high-level description of the SIMPL algorithm. Comprehensive details, as well as a theoretical analysis linking SIMPL formally to expectation-maximization of a class of generative models, is provided in the Appendix.

Algorithm 1 SIMPL: An algorithm for optimizing tuning curves and latents from behaviour

```
$\mathbf{s} \in \mathbb{N}^{N \times T}$                                        ▷ Spike counts
$\mathbf{x}^{(0)} \in \mathbb{R}^{D \times T}$                                  ▷ Initial latent estimate
procedure SIMPL($\mathbf{s}, \mathbf{x}^{(0)}$)
    for $e \leftarrow 0$ to $E$ do                                   ▷ Loop for $E$ iterations
        $\mathbf{f}^{(e)} \leftarrow$ FitTuningCurves($\mathbf{x}^{(e)}, \mathbf{s}$)                ▷ The "M-step"
        $\mathbf{x}^{(e+1)} \leftarrow$ DecodeLatent($\mathbf{f}^{(e)}, \mathbf{s}$)                ▷ The "E-step"
    end for
    return $\mathbf{x}^{(E+1)}, \mathbf{f}^{(E)}$                    ▷ The optimised latent and tuning curves
end procedure
```

2.1 THE MODEL

SIMPL models *spike trains* of the form $\mathbf{s} := (s_{it})_{t=1,\ldots T}^{i=1,\ldots N}$, where s_{it} represents the number of spikes emitted by neuron i between time $t \cdot \Delta t$ and $(t+1) \cdot \Delta t$, for some time discretization interval Δt. We denote $\mathbf{s}_t := (s_{1t}, \ldots, s_{Nt})$ the vector of spike counts emitted by all neurons in the t-th time bin. SIMPL posits that such spike trains $\mathbf{s}$ are modulated by a *latent, continuously-valued, time-evolving* variable $\mathbf{x} := (\mathbf{x}_t)_{t=1,\ldots,T} \in \mathbb{R}^D$ through the following random process:

$$\begin{aligned} s_{it} \mid \mathbf{x}_t &\sim \text{Poisson}(f_i(\mathbf{x}_t)) \\ \mathbf{x}_{t+1} \mid \mathbf{x}_t &\sim \mathcal{N}(\mathbf{x}_t, (\mathrm{v} \cdot \Delta t)^2 I), \end{aligned}$$

[1] Code and a demo can be found at: https://anonymous.4open.science/r/simpl/README.md

and $\mathbf{x}_0 \sim \mathcal{N}(0, \sigma_0 I)$. Here, v is some constant velocity hyperparameter. The resulting prior distribution of $\mathbf{x}$, (called $p_{\mathbf{x}}$) enforces temporal smoothness in the trajectories. The latent variable $\mathbf{x}_t$ determines the instantaneous firing rate of each neuron via its intensity function f_i (hereon called its *tuning curve*, collectively denoted $\mathbf{f}$), which is unknown a priori, and which SIMPL will estimate. Moreover, we make the common assumption that all neurons are *conditionally independent* given $\mathbf{x}_t$, i.e. $p(\mathbf{s}_t|\mathbf{x}_t) = \prod_{i=1}^{N} p(\mathbf{s}_{it}|\mathbf{x}_t)$. Finally, we assume the latent variable $\mathbf{x}$ evolves only according to its previous state (it is Markovian), a common assumption in the neuroscience literature (see, e.g. George et al. 2021). This model has been previously studied in the literature (Smith & Brown, 2003; Macke et al., 2011), albeit using restrictive intensity function models, something which SIMPL avoids as discussed below.

2.2 THE SIMPL ALGORITHM

Outline We now seek an estimate of the true, unknown latent trajectory $\mathbf{x}^\star$ and tuning curves $\mathbf{f}^\star$ that led to some observed spike train, $\mathbf{s}$. SIMPL does so by iterating a two-step procedure closely related to the expectation-maximisation (EM) algorithm: first, tuning curves are fitted to an initial estimate of the latent variable (the “M-step”), which are then used to decode the latent variable (the “E-step”). This procedure is then repeated using the new latent trajectory, and so on until convergence.

The M-step In the M-step of the e-th iteration (or “epoch”) given the current latent trajectory estimate $\mathbf{x}^{(e)}$, SIMPL fits intensity functions using kernel density estimation (KDE):

$$f_i^{(e)}(\mathbf{x}) := \frac{\sum_{t=1}^{T} s_{it}\, k(\mathbf{x}, \mathbf{x}_t^{(e)})}{\sum_{t=1}^{T} k(\mathbf{x}, \mathbf{x}_t^{(e)})} \approx \underbrace{\frac{\text{\# spikes at x}}{\text{\# visits to x}}}_{\text{undefined outside } \mathbf{x}^{(e)}} \tag{1}$$

The use of a kernel allows to extrapolate the intuitive estimate on the right of Equation 1 to locations x not present in the trajectory $\mathbf{x}^{(e)}$. In practice, we use a Gaussian kernel with bandwidth σ.

The E-step In the E (or decoding) step, SIMPL produces a new estimate $\mathbf{x}^{(e+1)}$ of the latent trajectory by smoothing across time the (non-smooth) maximum likelihood estimate of $\mathbf{x}$, given $\mathbf{s}$ and $\mathbf{f}^{(e)}$, using $p_{\mathbf{x}}$, the prior distribution on $\mathbf{x}$. In particular, a theoretical argument detailed in the appendix allows SIMPL to employ *Kalman Smoothing*, resulting in a principled and efficient decoding procedure, which we summarize below. The intuition is that although spike counts are *not* Gaussian, with a sufficient number of neurons the maximum likelihood estimate is Gaussian around $\mathbf{x}$ and so can be Kalman smoothed:

$$\begin{aligned} \widehat{\mathbf{x}}_t^{(e+1)} &:= \arg\max_{\mathbf{x}} \log p(\mathbf{s}_t|\mathbf{x}, \mathbf{f}^{(e)}) \\ \mathbf{x}_t^{(e+1)} &:= \text{KalmanSmooth}(\widehat{\mathbf{x}}_t^{(e+1)}, p_{\mathbf{x}}) \end{aligned} \tag{2}$$

Behavioural initialization Spike trains often come alongside behavioural recordings which are thought to be closely related to the latent variable $\mathbf{x}$. SIMPL leverages this by setting $\mathbf{x}^{(0)}$, the initial decoded latent trajectory, to measured behaviour. We posit that such *behavioural initialization* will place the first iterate of SIMPL in the vicinity of the true trajectory and tuning curves. This, in turn, faciliates the search for a good model, and favours the true latent and tuning curves $(\mathbf{x}^\star, \mathbf{f}^\star)$ over alternative pairs $(\phi(\mathbf{x}^\star), \mathbf{f}^\star \circ \phi^{-1})$ whose latent space is *warped* by some invertible map ϕ, and which would explain the data equally well. Through ablation studies, we confirm the beneficial effects of this behavior-informed initialization in the experiments section (see Fig. 3 and 4). To reinforce this incentive and further improve numerical stability, we also transform the decoded latent trajectory at each iteration using an linear map which maximally aligns it with behavior.

All in all, SIMPL is interpretable and closely matches common practice in neuroscience; moreover, it can be formally related to a generalized version of the EM-algorithm, for which theoretical guarantees may be obtained under suitable assumptions. We describe in detail the theoretical arguments justifying the validity of SIMPL as well as its connection to EM in the appendix.

3 RELATED WORK

Probabilistic inference in spike trains modulated by latent variables has been a major topic in neural data analysis for decades — see, e.g. Yu et al. (2005; 2006; 2008b;a); Macke et al. (2011); Mangion et al. (2011); Park et al. (2015); Gao et al. (2016); Duncker et al. (2019); Zhou & Wei (2020); Schneider et al. (2023). Closest to SIMPL are the works of Smith & Brown (2003); Macke et al. (2011), which both perform approximate EM in a hidden markov model with Poisson emissions and a Gaussian random walk prior on $\mathbf{x}$. Both methods use a simplistic parametric linear–exponential model of intensity functions; such parametric models are not flexible enough to capture neurons with complex tuning properties such as place cells and grid cells. The (approximate) E-step of Macke et al. (2011) employs a global Laplace approximation, leveraging the concavity of the log-posterior of such models to compute the maximum a posteriori (MAP) of the entire trajectory; however, this concavity is a consequence of the intensity function model, and does not hold in our case. On the other hand, the approximate E-step of Smith & Brown (2003) uses a *local* Laplace approximation to obtain the MAP. However, their algorithm requires running optimization algorithms sequentially, which can be computationally expensive. In contrast, the MLE optimization problems computed in SIMPL's E-step can be solved in parallel across time points, making SIMPL more scalable.

Markovian models assume that the future trajectory of an agent is only influenced by its current state, not its past ones. This assumption may not accurately capture certain brain patterns with long range time dependencies. To address such issues, a series of methods, pioneered by Yu et al. (2008a), and refined in Wu et al. (2017); Zhao & Park (2017); Jensen et al. (2020) instead consider spike train models using a Gaussian Process prior on $\mathbf{x}$, which only enforces smoothness in the latent dynamics. However, inference using Gaussian processes is computationally expensive in the number of time points, requiring additional approximations to remain tractable thus these techniques are typically used for very short neural datasets unlike the $\mathcal{O}$(hours)-long datasets we consider here.

To model complex non-linear, but Markovian, transition structures and alleviate some time scaling issues of GP methods, LFADS (Pandarinath et al., 2018) uses a Recurrent Neural Network to model latent dynamics. While LFADS is capable of modelling a wide range of firing patterns, its linear–exponential intensity function model will, again, not capture the complex tuning properties of grid cells and place cells. Moreover, LFADS comes with expensive training overheads and hyperparameters which are reportedly hard to tune (Keshtkaran et al., 2022). Pi-VAE (Zhou & Wei, 2020) uses a Variational Autoencoder (Kingma & Welling, 2014) to learn both a generative model and a latent decoding network for latent-modulated spike events. Finally, CEBRA (Schneider et al., 2023) is a neural network based technique that learns a deterministic encoder mapping spikes to latents using Noise–Contrastive Estimation. CEBRA focuses on decoding and does not natively learn intensity functions, which are of primary interest in our setting.

4 RESULTS

4.1 TOY MODEL OF A DISCRETE LATENT VARIABLE TASK

Before testing SIMPL on a large temporally continuous dataset we constructed a smaller dataset akin to a discrete two-alternative forced choice task (2AFC) (Fig. 2) — a widely studied decision–making paradigm (Platt & Glimcher, 1999; Bogacz et al., 2006; Znamenskiy & Zador, 2013; Lieder et al., 2019). The true latent states $\mathbf{x}_t^\star \in \{0, 1\}$ are binary and have no temporal structure (here subscript t indexes *trials* not time), analagous to a series of random "left" or "right" choices (Fig. 2b). This latent state is stochastically encoded by a population of neurons with random tuning curves giving the Bernoulli emission probabilities under each latent state:

$$f_i^\star(\mathbf{x}) = \begin{cases} f_{i0} \sim \mathcal{U}(0,1) & \mathbf{x} = 0, \\ f_{i1} \sim \mathcal{U}(0,1) & \mathbf{x} = 1, \end{cases}$$

$$\mathbf{x}_t^\star \sim \text{Bernoulli}(0.5) \quad \text{and} \quad s_{it}|\mathbf{x}_t \sim \text{Bernoulli}(f_i^\star(x_t^\star)).$$

Data is then sampled for $T = 50$ trials and $N = 15$ neurons as shown in Fig. 2. Initial conditions, $\mathbf{x}_t^{(0)}$, are generated from the true latent by randomly resampling a fraction of trials $\rho = 0.5$ (Fig. 2b). This partial resample represents an initial discrepancy between the behavioural measurement and the true internal state of the agent. We perform inference on this dataset using a reduced version of the model (SIMPL-R). In the M-step, tuning curves were simple fitted by calculating the average activity of a neuron across each latent condition (e.g. $f_i^{(e)}(\mathbf{x}) = \sum_t s_{it}\delta(\mathbf{x}_t^{(e)}, \mathbf{x}) / \sum_t \delta(\mathbf{x}_t^{(e)}, \mathbf{x})$, conceptually similar to KDE). For the E-step, each latent was the decoded according to the maximum likelihood estimate under the observed spikes and tuning curve estimates from the previous

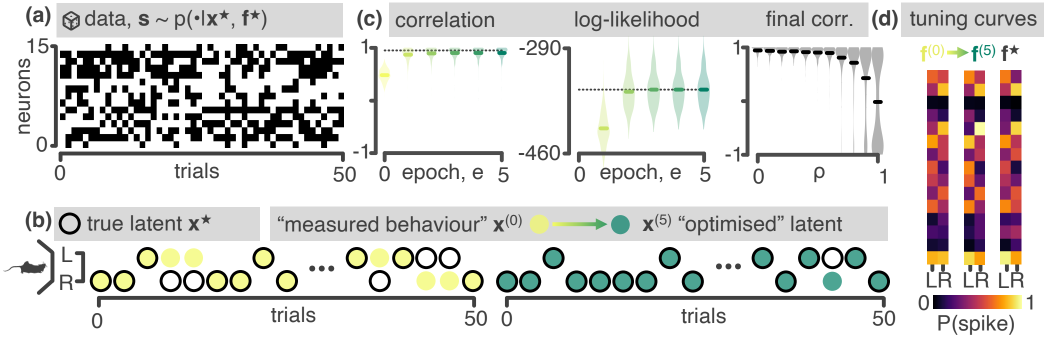

Figure 2: A two-alternative forced choice task (2AFC) toy-model. **(a)** Data generation: Spikes are sampled from a simple generative model. For each of T=50 independent trials a random binary latent — analogous to a "left" or "right" choice — is encoded by a population of N=15 neurons with randomly initialized tuning curves. **(b)** Model performance: Starting from a noisy estimate (yellow) of the true latent (black) where a fraction $\rho = 0.5$ of trials are resampled, SIMPL-R recovers the true latent variables (green) with high accuracy. **(c)** *Left:* Correlation between $\mathbf{x}^{(e)}$ and $\mathbf{x}^{\star}$. *Middle:* Log-likelihood, $\log p(\mathbf{s}|\mathbf{x}^{(e)}, \mathbf{f}^{(e)})$. *Right:* Final correlation between $\mathbf{x}^{(5)}$ and $\mathbf{x}^{\star}$ as a function of initialization noise ρ. Violin plots show distributions over 1000 randomly seeded datasets, dotted lines show ceiling performance of a perfectly initialized model ($\mathbf{x}^{(0)} = \mathbf{x}^{\star}$) **(d)** Tuning curves.

epoch: $\mathbf{x}_t^{(e+1)} = \arg\max_{\mathbf{x}} \sum_i \log p(s_{it}|\mathbf{x}, f_i^{(e)})$ (there is no time dependence between latents, thus no Kalman smoothing). This process was repeated for 5 epochs and, with high reliability, converged on the true latents after approximately two (Fig. 2c & d, distributions show repeat for 1000 randomly seeded datasets, dotted lines show ceiling performance on a model perfectly initialized with noiseless $\mathbf{x}^{(0)} = \mathbf{x}^{\star}$). We repeated this experiment for various values of ρ: latent recovery was almost perfect when ρ was small (i.e. when the initial conditions were close to the true latent), dropping off as ρ approached 1. At $\rho = 1$ when the conditions were *completely* random, the model was biased to recover a latent space that is either perfectly correlated or perfectly anti-correlated ("left" ↔ "right") with the true latent (Fig. 2c, right), a valid isomorphism discussed more in the upcoming section.

4.2 CONTINUOUS SYNTHETIC DATA: 2D GRID CELLS

Next we tested SIMPL on a realistic navigational task by generating a large artificial dataset of spikes from a population of $N = 225$ 2D grid cells — a type of neuron commonly found in the medial entorhinal cortex which activate on the vertices of a regular hexagonal grid (Hafting et al., 2005) — in a 1 m square environment. Grid cell tuning curves, $\mathbf{f}^{\star}$, were modelled as the thresholded sum of three planar waves at 0°, 60° and 120° to some offset direction (a commonly used model within the computational neuroscience literature (George et al., 2024a)) and, as observed in the brain, cells were arranged into three discrete modules, 75 cells per modules, of increasing grid scale from 0.3–0.8 m (Fig. 3c). Each cell had a maximum firing rate of 10 Hz. A latent trajectory, $\mathbf{x}^{\star}$, was then generated by simulating an agent moving around the environment for 1 hour under a smooth continuous random motion model which had been fitted to rodent foraging behaviour. Data was sampled at a rate of 10 Hz giving a total of $T = 36,000$ time bins ($\sim$ 800,000 spikes). All data was generated using the `RatInABox` (George et al., 2024a).

$$\mathbf{x}^{\star} \sim \text{Smooth-continuous-random-walk} \quad \text{and} \quad s_{it}|\mathbf{x}_t^{\star} \sim \text{Poi}(\mathbf{f}_i^{\text{GC}}(\mathbf{x}_t^{\star})) \tag{3}$$

The initial latent trajectory, $\mathbf{x}^{(0)}$, was generated by adding smooth Gaussian noise to the latent $\mathbf{x}$ such that, on average, the true latent and initial condition differed by 20 cm (Fig. 3a, top panel). This discrepancy, potentially representing the agents internal uncertainty in their position, was sufficient to obscure almost all structure from the grid cell tuning curves $\mathbf{f}^{(0)}(\mathbf{x})$ for all but the largest grid scales (Fig. 3b, top).

To assess performance we partition the spike data matrix, $\mathbf{s}$, into testing and training sets, $\mathcal{S}_{\text{test}}, \mathcal{S}_{\text{train}}$: inference is performed exclusively using data in the training set, and we then track the log-likelihood of data in both sets (Fig. 3d, left), e.g. $\ell_{\text{test}}^{(e)} = |\mathcal{S}|_{\text{test}}^{-1} \sum_{(i,t)\sim\mathcal{S}^{\text{test}}} \log p(s_{it}|\mathbf{x}_t^{(e)}, \mathbf{f}_i^{(e)})$. This partitioning has to be done with care; entire time intervals cannot be withheld for testing without impairing

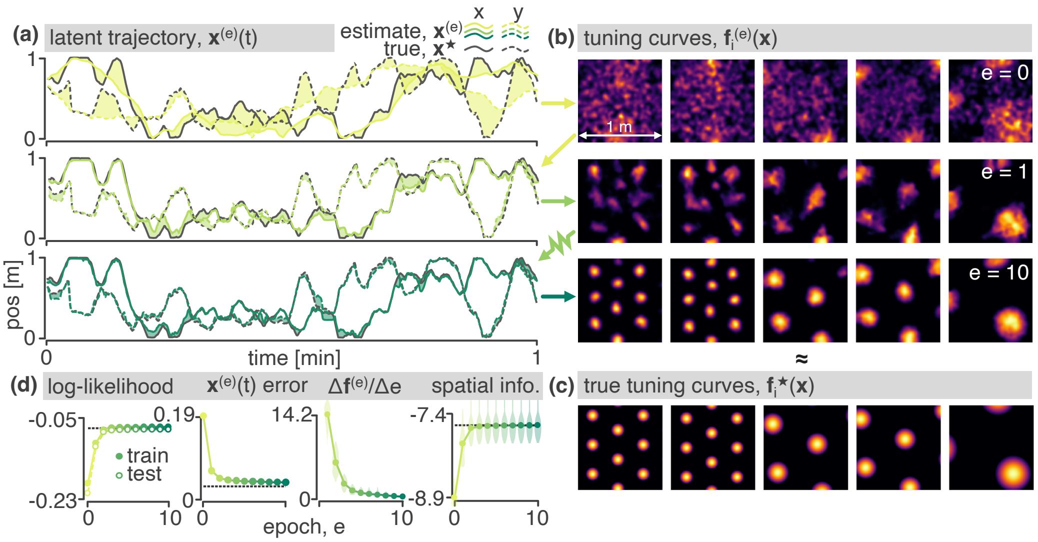

Figure 3: Results on a synthetic 2D grid cell dataset. An artificial agent locomotes a 1 m square environment for 1 hour (Δt = 0.1 s). Spikes are generated from N=225 artificial grid cells. **(a)** Estimated latent trajectories shown for epochs 0, 1 and 10. x and y positions are denoted by dotted and dashed lines respectively. Initial conditions are generated from the true latent (black) by the addition of smooth continuous Gaussian noise. **(b)** Tuning curve estimates for 5 examplar grid cells at epochs 0, 1 and 10. **(c)** Ground truth tuning curves. **(d)** Performance metrics: *Left:* log-likelihood of the train and test spikes (averaged per time step, dotted line shows ceiling performance on a model initialized with the true latent). *Middle-left:* Euclidean distance between the true and estimated latent trajectories (averaged per time step). *Middle-right:* Epoch-to-epoch change in the tuning curves. *Right:* Cell spatial information. Violin plots, where shown, display distributions across all 225 neurons.

the model's ability to infer the latent over this period. Likewise, entire neurons cannot be withheld without impairing the model's capacity to estimate their tuning curves. Instead, we adopt a speckled train-test mask previously used in latent variable modelling set-ups (Williams et al., 2020) which withholds for testing extended chunks of time bins arranged in an irregular "speckled" pattern across the data matrix (totalling 10% of the data) without ever removing all neurons for a given time bin or all time bins for a given neuron. We also calculate the Euclidean distance between the true and latent trajectory (Fig. 3d, middle-left), $T^{-1}\sum_t \|\mathbf{x}^{(e)}(t) - \mathbf{x}_t\|_2$, the epoch-to-epoch change in the tuning curves (Fig. 3d, middle-right) and the entropy (hereon called "spatial info", Fig. 3d, right) of the normalized tuning curves as a measure of how spatially informative they are.

SIMPL was then run for 10 epochs (total compute time 39.8 CPU-secs). The true latent trajectory and receptive fields were recovered almost perfectly and the log-likelihood of both train and test spikes rapidly approached the ceiling performance with only slight overfitting.

Influence of behavioural initializations on performance Latent variable models trained with EM can experience two issues that usually complicate the scientific interpretability of their results. First, they may not converge to a good model of the data; second, even if they do, the recovered latent spaces and intensity functions $(\mathbf{f}^{(e)}, \mathbf{x}^{(e)})$ may differ from the true ones $(\mathbf{f}^{\star}, \mathbf{x}^{\star})$ by some invertible "warp" ϕ that does not affect the overall goodness of fit of the model. While SIMPL is a latent variable model, we show that behavioural initialization drastically minimizes the severity of these issues. To do so, we first assess the absolute goodness–of–fit of SIMPL by computing the correlation between the estimated instantaneous firing rates $f^{(e)}(x_t^{(e)})$ (a quantity invariant to warping) and the true ones. Our analysis shows that SIMPL converges to a highly accurate model (r=0.98) under behavioural initialization, but to a less accurate (but still quite accurate) one (r = 0.87) when initialized with a random latent trajectory which is uncorrelated with behavior. Second, we estimate, quantify and visualize the warp map ϕ between SIMPL's estimates $(\mathbf{f}^{(e)}, \mathbf{x}^{(e)})$ and the ground truth $(\mathbf{f}^{\star}, \mathbf{x}^{\star})$. We obtain this estimate by finding a mapping from the discovered latent space to the true latent space which minimizes the L2 difference between the tuning curves ($\phi(\mathbf{x})$ =

$\arg\min_{\mathbf{y}} \|\mathbf{f}^{\star}(\mathbf{y}) - \mathbf{f}^{(e)}(\mathbf{x})\|_2)$. We then quantify the "warpness" of this mapping by calculating the average distance between $\mathbf{x}$ and $\phi(\mathbf{x})$ across the environment, normalized by its characteristic length scale (1 m). This warp distance should be 0 for total un-warped models and $\mathcal{O}(1)$ for heavy warps. We find that in addition to perfectly fitting the data, the solution found by SIMPL under behavioural initialization is minimally warped (warp dist = 0.050). In contrast, the good (but imperfect) solution found by SIMPL under random initialization is heavily warped (warp dist. = 0.498) in a fragmented manner. These results are shown in Fig. 4 and strongly motivate the use of behavioural initializations in latent variable models as an effective mean to encourage convergence towards latent spaces which are both accurate and un-warped with respect to the ground truth.

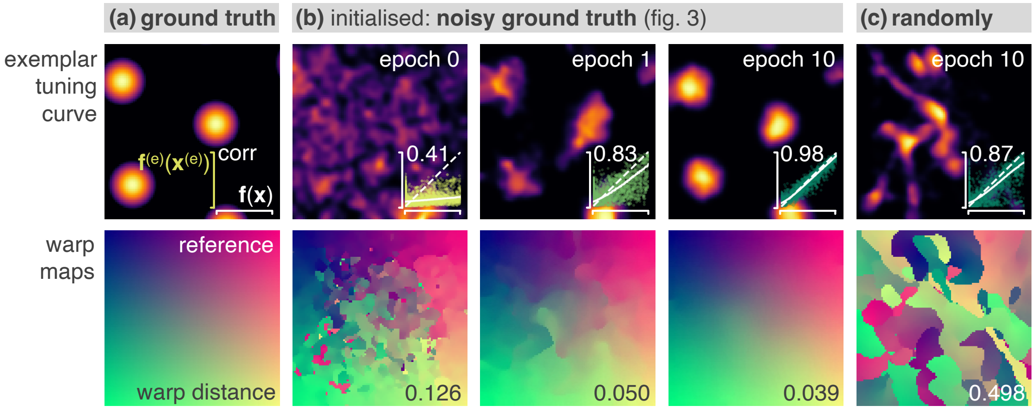

Figure 4: Latent manifold analysis: **(Top)** Examplar tuning curve in the ground truth latent space **(a)**, the latent space discovered by behaviourally-initialised-SIMPL after 0, 1 and 10 epochs **(b)** and the latent space discovered by SIMPL initialized with a random latent trajectory **(c)**. Inset scatter plots show the true and predicted firing rates of all neurons across all times as well as their correlation values ("accurate" models have higher correlations). **(Bottom)** Visualizations of the warp functions mapping each latent space to the "closest" location in ground truth as measured by the distance between the tuning curves population vectors.

Comparison to CEBRA We compared SIMPL to a popular latent variable extraction technique called CEBRA (Schneider et al., 2023). Unline SIMPL which uses behaviour as an initialisation, CEBRA learns latent embedding directly from spikes by training a deep neural network to minimise a contrastive loss function with behaviour as the labels. We trained CEBRA on our synthetic grid cell data using out-of-the-box hyperparameters[2] training for the default 10000 iterations. After training we aligned the latent to behaviour and observed that CEBRA found a latent trajectory (Fig. 5a, blue) very close to the true latent (Fig. 5a, black) much like SIMPL. CEBRA's latent embedding was noisier than SIMPL (a likely consequence of the explicit smoothing we perform) and had significantly larger final error (9.2 cm vs 4.0 cm). Since CEBRA doesn't explicitly learn a generative model in order to visualise tuning curves we we applied our standard KDE fitting procedure (an M-step) to the CEBRA latents. The resulting grid cells but remained blurry relative to the ground truth (but were better than behaviour), in comparison to SIMPL, which produced sharp, well-defined grid fields (Fig. 5b) close to the ground truth. CEBRA took just over 23 minutes to train on a consumer laptop with 8-CPUs compared to just under 40 seconds for SIMPL on the same machine.

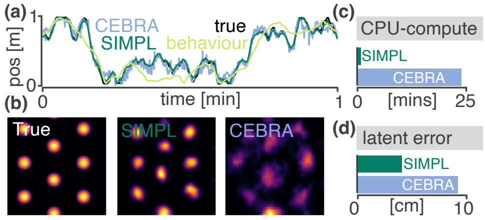

Figure 5: Comparison between SIMPL and CEBRA.

[2] with the exception that we turned 'off' normalisation so outputs weren't normalised onto a sphere

4.3 Hippocampal place cell data

Finally, we test SIMPL on a neural dataset from $N = 226$ hippocampal neurons recorded from a rat as it foraged in a large 3.5 m by 2.5 m environment for 2 hours (full details can be found in Tanni et al. 2022). The data was binned at 5 Hz ($\mathrm{d}t = 0.2s$ giving $T = 36,000$ data samples, total $\sim$ 700,000 spikes). Place cells are a type of neuron commonly found in the hippocampus which activate when an animal is in a specific location in space (its "place field") and, like grid cells, are thought to be a key component of the brain's navigational system (O'Keefe, 1978). In large environments place cells are known to exhibit tuning curves with multiple place fields (Park et al., 2011).

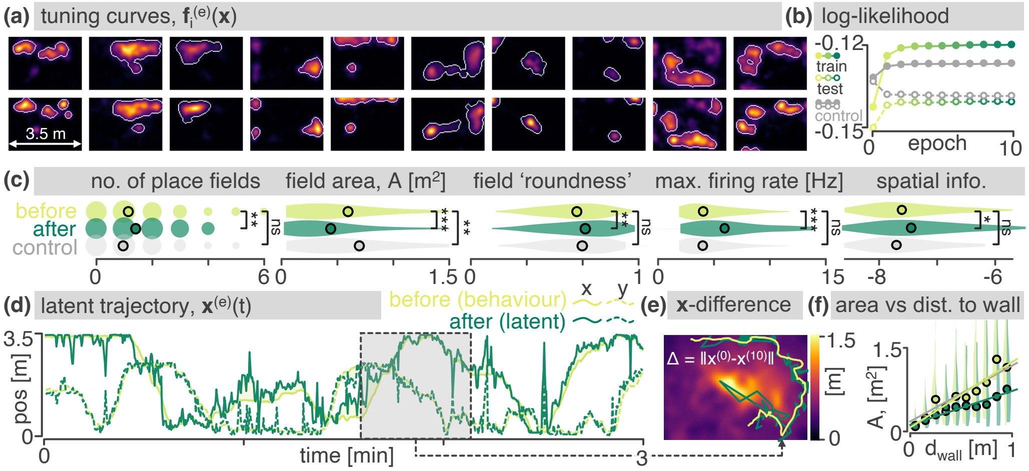

Figure 6: Results on a hippocampal place cell dataset collected by Tanni et al. (2022). **(a)** Exemplar tuning curves before and after optimization. Automatically identified place field boundaries shown in white. **(b)** Log-likelihood of test and train spikes. Equivalent results for a control model — fitted with spikes resampled from the behavioural place fields, $\mathbf{s}_{\text{control}} \sim p(\cdot|\mathbf{x}^{(0)}, \mathbf{f}^{(0)})$ — shown in grey. **(c)** Place field (before, after, control-after) analysis. Violin plots show the distributions over all place fields / place cells. **(d)** The final latent trajectory estimated from SIMPL (green) overlaid on top of the behaviour (used as initial conditions) (yellow). x and y coordinates shown with dotted and dashed lines respectively. **(e)** Behavioural discrepancy map: the average discrepancy $\|\mathbf{x}_t^{(0)} - \mathbf{x}_t^{(10)}\|_2$ as a function of the optimized latent $\mathbf{x}^{(10)}$. Overlaid is a snippet of the behavioural vs optimized true latent trajectory. **(f)** Median place field sizes, and distributions, as a function of the distance to the nearest.

We initialized SIMPL using the measured position of the animal and optimized for 10 epochs. The log-likelihood of test and train spikes increased, Fig. 6b, converging after approximately 4 epochs (compute time 41.2 CPU-secs). Place fields were automatically identified by thresholding the activity of each neuron at 1 Hz and identifying contiguous regions of activity with a peak firing rate above 2 Hz and a total area less than half that of the full environment, similar to previous work (Tanni et al., 2022).

Tuning curves were visibly sharper after optimization, Fig. 6a; diffuse place fields shrunk (e.g. see the third exemplar tuning curve) or split into multiple, smaller fields (second exemplar) (Fig. 6a). Occasionally, new place fields appeared (fourth exemplar) or multiple place fields merged into a single larger field (fifth exemplar). Statistically, tuning curves had significantly more individual place fields (mean 1.14→1.41 per cell, $p = 0.0035$ Mann Whitney U tests), substantially higher maximum firing rates (median 4.2→6.1 Hz, $p = 9.8 \times 10^{-7}$) and were more spatially informative ($p = 0.038$). Individual place fields were substantially smaller (median 0.59→0.44 m^2) and rounder (median 0.63→0.68, $p = 0.0037$). Notably only *place* cells — defined as cells with at least one place field — showed significant changes in their tuning curves, non-place cells were statistically unaffected (data not shown).

To ensure that these changes were not an artefact of the SIMPL algorithm we generated a control dataset by resampling spikes from the behaviour-fitted tuning curves, $\mathbf{s}_{\text{control}} \sim p(\cdot|\mathbf{x}^{(0)}, \mathbf{f}^{(0)})$. Control spikes thus had very similar temporal statistics and identical tuning curves to those in the original dataset but, crucially, were generated from a known ground truth model exactly equal to the initialization. Thus, any changes to the tuning curves under SIMPL optimization can be considered artefactual and not fundamental to the underlying neural data. No significant effect of optimization on the control data (except for a slight *increase* in field area) was observed and all measured effects – though statistically insignificant – pointed in the *opposite* direction to those observed in the real data (except for roundness) (Fig. 6c). This control provides strong evidence that the changes observed in the real data are genuine and reflect the true nature of neural tuning curves in the brain.

After optimization the latent trajectory $\mathbf{x}^{(10)}$ remained highly correlated with the behaviour ($R^2 = 0.86$, fig. 6d) occasionally diverging for short period as the latent "jumped" to and from a new location, as if the animal was mentally teleporting itself (one such "jump" is visualized in Fig. 6e). The close correspondence between the optimized latent and the behaviour allows us to directly compare when, and where, they diverge. We calculated the discrepancy between the optimized latent and the behaviour at each time point, $\|\mathbf{x}_t^{(0)} - \mathbf{x}_t^{(10)}\|_2$, and visualized this as a heat map overlaid onto the latent space (Fig. 6e). Discrepancy was minimal around the edges of the environment and peaked near the centre, consistent with the hypothesis that sensory input is less reliable in the centre of the environment (where there are fewer visual and tactile cues) to guide self-localisation resulting in a larger average discrepancy between the optimized latent and the behaviour.

(Tanni et al., 2022) found that place field size increased with distance from the nearest wall in the environment. Our observation — that latent-behaviour discrepancy is highest in the centre of the environment — suggests a possible explanation: place fields in the centre of the environment *appear* larger because they are distorted by the discrepancy which is asymmetric across the environment. To test this we binned place fields according to their distance to the nearest wall (measured with respect to the place fields centre of mass) and plotted the median field size against distance (Fig. 6f). Optimized place fields, much like behavioural place fields, were the smallest near the walls and grew with distance (replicating (Tanni et al., 2022)), but this correspondence broke down around ~ 0.5 m after which the optimized size distribution flattened off, something not observed in the control. A majority of the shrink in place field size thus came from larger place fields near the centre of the environment not the smaller ones near the walls. This result suggests that a substantial fraction of the increased size of place fields away from walls is not a fundamental feature of the neural tuning curves themselves but can be attributed to a behaviour-induced distortion in the tuning curves, an artefact which can be 'undone' by SIMPL.

5 DISCUSSION

We introduced SIMPL, a tool for optimizing tuning curves and latent trajectories using a technique which refines estimates obtained from behaviour. It hinges on two well-established sub-routines — fitting and decoding — which are widely used by both experimentalists and theorists for analysing neural data. By presenting SIMPL as an iterative application of these techniques, we aim to make latent variable modelling more accessible to the neuroscience community.

Furthermore, we see SIMPL as a specific instance of a broader class of latent optimization algorithms. In principle *any* arbitrary curve fitting procedure and *any* arbitrary decoder could be coupled into a candidate algorithm for optimizing latents from neural data. Our specific design choices, while attractive due to their conceptual simplicity, will also come with limitations. For example, we predict KDE won't scale well to very high dimensional latent spaces (Györfi et al., 2006). In these instances user could consider substituting this component with a parametric model such as neural network which are known to perform better in high dimensions (Bach, 2017), potentially at the cost of training time.

Our synthetic analysis focussed on settings where behaviour and the true latent differed only in an unbiased manner. It would be interesting to determine if SIMPL 's strong performance extends to more complex perturbations. In the brain, asymmetric perturbations are common; for instance, during theta sweeps (Maurer et al., 2006), the encoded latent moves away from the agent. This forward-biased discrepancy could theoretically induce a backward-biased skew in behavioral place fields, even if the true tuning curves remain unskewed. If this is the case, latent dynamics —– and tools like SIMPL for extracting them — could help reinterpret the predictive nature of place field tuning curves Stachenfeld et al. (2017); Fang et al. (2023); Bono et al. (2023); George et al. (2023), similar to how latent optimization reduced the asymmetry in place field sizes further from walls (Fig. 6f).

References

Afsheen Afshar, Gopal Santhanam, M Yu Byron, Stephen I Ryu, Maneesh Sahani, and Krishna V Shenoy. Single-trial neural correlates of arm movement preparation. *Neuron*, 2011.

Evan W Archer, Urs Koster, Jonathan W Pillow, and Jakob H Macke. Low-dimensional models of neural population activity in sensory cortical circuits. *Advances in neural information processing systems*, 2014.

Francis Bach. Breaking the curse of dimensionality with convex neural networks. *Journal of Machine Learning Research*, 18(19):1–53, 2017.

Patrick Billingsley. Statistical methods in markov chains. *The annals of mathematical statistics*, 1961.

Rafal Bogacz, Eric Brown, Jeff Moehlis, Philip Holmes, and Jonathan D Cohen. The physics of optimal decision making: a formal analysis of models of performance in two-alternative forced-choice tasks. *Psychological review*, 2006.

Jacopo Bono, Sara Zannone, Victor Pedrosa, and Claudia Clopath. Learning predictive cognitive maps with spiking neurons during behavior and replays. *Elife*, 12:e80671, 2023.

James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. JAX: composable transformations of Python+NumPy programs, 2018. URL `http://github.com/jax-ml/jax`.

Ralph A Bradley and John J Gart. The asymptotic properties of ml estimators when sampling from associated populations. *Biometrika*, 1962.

Federico Carnevale, Victor de Lafuente, Ranulfo Romo, Omri Barak, and Néstor Parga. Dynamic control of response criterion in premotor cortex during perceptual detection under temporal uncertainty. *Neuron*, 2015.

Margaret F Carr, Shantanu P Jadhav, and Loren M Frank. Hippocampal replay in the awake state: a potential substrate for memory consolidation and retrieval. *Nature neuroscience*, 2011.

John P Cunningham and Byron M Yu. Dimensionality reduction for large-scale neural recordings. *Nature neuroscience*, 2014.

Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society: series B (methodological)*, 1977.

Christian F Doeller, Caswell Barry, and Neil Burgess. Evidence for grid cells in a human memory network. *Nature*, 2010.

Lea Duncker, Gergo Bohner, Julien Boussard, and Maneesh Sahani. Learning interpretable continuous-time models of latent stochastic dynamical systems. In *International conference on machine learning*. PMLR, 2019.

Ching Fang, Dmitriy Aronov, LF Abbott, and Emily L Mackevicius. Neural learning rules for generating flexible predictions and computing the successor representation. *elife*, 12:e80680, 2023.

André A Fenton and Robert U Muller. Place cell discharge is extremely variable during individual passes of the rat through the firing field. *Proceedings of the National Academy of Sciences*, 1998.

Ronald Aylmer Fisher. Theory of statistical estimation. In *Mathematical proceedings of the Cambridge philosophical society*. Cambridge University Press, 1925.

Yuanjun Gao, Evan W Archer, Liam Paninski, and John P Cunningham. Linear dynamical neural population models through nonlinear embeddings. *Advances in neural information processing systems*, 2016.

Dileep George, Rajeev V Rikhye, Nishad Gothoskar, J Swaroop Guntupalli, Antoine Dedieu, and Miguel Lázaro-Gredilla. Clone-structured graph representations enable flexible learning and vicarious evaluation of cognitive maps. *Nature communications*, 2021.

Tom M George, William de Cothi, Kimberly L Stachenfeld, and Caswell Barry. Rapid learning of predictive maps with stdp and theta phase precession. *Elife*, 12:e80663, 2023.

Tom M George, Mehul Rastogi, William de Cothi, Claudia Clopath, Kimberly Stachenfeld, and Caswell Barry. Ratinabox, a toolkit for modelling locomotion and neuronal activity in continuous environments. *Elife*, 2024a.

Tom M George, Kimberly L Stachenfeld, Caswell Barry, Claudia Clopath, and Tomoki Fukai. A generative model of the hippocampal formation trained with theta driven local learning rules. *Advances in Neural Information Processing Systems*, 2024b.

László Györfi, Michael Kohler, Adam Krzyzak, and Harro Walk. *A distribution-free theory of nonparametric regression*. Springer Science & Business Media, 2006.

Torkel Hafting, Marianne Fyhn, Sturla Molden, May-Britt Moser, and Edvard I Moser. Microstructure of a spatial map in the entorhinal cortex. *Nature*, 2005.

Christopher D Harvey, Philip Coen, and David W Tank. Choice-specific sequences in parietal cortex during a virtual-navigation decision task. *Nature*, 2012.

Pierre Hodara, Nathalie Krell, and Eva Löcherbach. Non-parametric estimation of the spiking rate in systems of interacting neurons. *Statistical Inference for Stochastic Processes*, 2018.

Øyvind Arne Høydal, Emilie Ranheim Skytøen, Sebastian Ola Andersson, May-Britt Moser, and Edvard I Moser. Object-vector coding in the medial entorhinal cortex. *Nature*, 2019.

Aapo Hyvärinen and Petteri Pajunen. Nonlinear independent component analysis: Existence and uniqueness results. *Neural networks*, 1999.

Kristopher Jensen, Ta-Chu Kao, Marco Tripodi, and Guillaume Hennequin. Manifold gplvms for discovering non-euclidean latent structure in neural data. *Advances in Neural Information Processing Systems*, 2020.

James J Jun, Nicholas A Steinmetz, Joshua H Siegle, Daniel J Denman, Marius Bauza, Brian Barbarits, Albert K Lee, Costas A Anastassiou, Alexandru Andrei, cite Çağatay Aydın, et al. Fully integrated silicon probes for high-density recording of neural activity. *Nature*, 2017.

Mohammad Reza Keshtkaran, Andrew R Sedler, Raeed H Chowdhury, Raghav Tandon, Diya Basrai, Sarah L Nguyen, Hansem Sohn, Mehrdad Jazayeri, Lee E Miller, and Chethan Pandarinath. A large-scale neural network training framework for generalized estimation of single-trial population dynamics. *Nature Methods*, 2022.

Diederik P. Kingma and Max Welling. Auto-Encoding Variational Bayes. In *2nd International Conference on Learning Representations, ICLR*, 2014.

Dmitry Kobak, Wieland Brendel, Christos Constantinidis, Claudia E Feierstein, Adam Kepecs, Zachary F Mainen, Xue-Lian Qi, Ranulfo Romo, Naoshige Uchida, and Christian K Machens. Demixed principal component analysis of neural population data. *elife*, 2016.

Colin Lever, Stephen Burton, Ali Jeewajee, John O'Keefe, and Neil Burgess. Boundary vector cells in the subiculum of the hippocampal formation. *Journal of Neuroscience*, 2009.

Itay Lieder, Vincent Adam, Or Frenkel, Sagi Jaffe-Dax, Maneesh Sahani, and Merav Ahissar. Perceptual bias reveals slow-updating in autism and fast-forgetting in dyslexia. *Nature neuroscience*, 2019.

Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Raetsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. In *international conference on machine learning*, 2019.

Ryan J Low, Sam Lewallen, Dmitriy Aronov, Rhino Nevers, and David W Tank. Probing variability in a cognitive map using manifold inference from neural dynamics. *BioRxiv*, 2018.

Jakob H Macke, Lars Buesing, John P Cunningham, Byron M Yu, Krishna V Shenoy, and Maneesh Sahani. Empirical models of spiking in neural populations. *Advances in neural information processing systems*, 2011.

Emily L Mackevicius, Andrew H Bahle, Alex H Williams, Shijie Gu, Natalia I Denisenko, Mark S Goldman, and Michale S Fee. Unsupervised discovery of temporal sequences in high-dimensional datasets, with applications to neuroscience. *Elife*, 2019.

Andrew Zammit Mangion, Ke Yuan, Visakan Kadirkamanathan, Mahesan Niranjan, and Guido Sanguinetti. Online variational inference for state-space models with point-process observations. *Neural computation*, 2011.

Valerio Mante, David Sussillo, Krishna V Shenoy, and William T Newsome. Context-dependent computation by recurrent dynamics in prefrontal cortex. *nature*, 2013.

Andrew P Maurer, Stephen L Cowen, Sara N Burke, Carol A Barnes, and Bruce L McNaughton. Organization of hippocampal cell assemblies based on theta phase precession. *Hippocampus*, 2006.

Bruce L McNaughton, Carol A Barnes, and JJEBR O'Keefe. The contributions of position, direction, and velocity to single unit activity in the hippocampus of freely-moving rats. *Experimental brain research*, 1983.

Mayank R Mehta, Carol A Barnes, and Bruce L McNaughton. Experience-dependent, asymmetric expansion of hippocampal place fields. *Proceedings of the National Academy of Sciences*, 1997.

May-Britt Moser, David C Rowland, and Edvard I Moser. Place cells, grid cells, and memory. *Cold Spring Harbor perspectives in biology*, 2015.

Robert U Muller and John L Kubie. The firing of hippocampal place cells predicts the future position of freely moving rats. *Journal of Neuroscience*, 1989.

J O'Keefe. The hippocampus as a cognitive map, 1978.

John O'Keefe and Jonathan Dostrovsky. The hippocampus as a spatial map: preliminary evidence from unit activity in the freely-moving rat. *Brain research*, 1971.

Chethan Pandarinath, Daniel J O'Shea, Jasmine Collins, Rafal Jozefowicz, Sergey D Stavisky, Jonathan C Kao, Eric M Trautmann, Matthew T Kaufman, Stephen I Ryu, Leigh R Hochberg, et al. Inferring single-trial neural population dynamics using sequential auto-encoders. *Nature methods*, 2018.

EunHye Park, Dino Dvorak, and André A Fenton. Ensemble place codes in hippocampus: Ca1, ca3, and dentate gyrus place cells have multiple place fields in large environments. *PloS one*, 2011.

Mijung Park, Gergo Bohner, and Jakob H Macke. Unlocking neural population non-stationarities using hierarchical dynamics models. *Advances in Neural Information Processing Systems*, 2015.

Michael L Platt and Paul W Glimcher. Neural correlates of decision variables in parietal cortex. *Nature*, 1999.

Herbert E Rauch, F Tung, and Charlotte T Striebel. Maximum likelihood estimates of linear dynamic systems. *AIAA journal*, 1965.

Honi Sanders, César Rennó-Costa, Marco Idiart, and John Lisman. Grid cells and place cells: an integrated view of their navigational and memory function. *Trends in neurosciences*, 2015.

Steffen Schneider, Jin Hwa Lee, and Mackenzie Weygandt Mathis. Learnable latent embeddings for joint behavioural and neural analysis. *Nature*, 2023.

Anne C Smith and Emery N Brown. Estimating a state-space model from point process observations. *Neural computation*, 2003.

Hugo J Spiers and Eleanor A Maguire. Thoughts, behaviour, and brain dynamics during navigation in the real world. *Neuroimage*, 2006.

Larry R Squire, Anna S van der Horst, Susan GR McDuff, Jennifer C Frascino, Ramona O Hopkins, and Kristin N Mauldin. Role of the hippocampus in remembering the past and imagining the future. *Proceedings of the National Academy of Sciences*, 2010.

Kimberly L Stachenfeld, Matthew M Botvinick, and Samuel J Gershman. The hippocampus as a predictive map. *Nature neuroscience*, 2017.

Sander Tanni, William De Cothi, and Caswell Barry. State transitions in the statistically stable place cell population correspond to rate of perceptual change. *Current Biology*, 2022.

Jeffrey S Taube, Robert U Muller, and James B Ranck. Head-direction cells recorded from the post-subiculum in freely moving rats. i. description and quantitative analysis. *Journal of Neuroscience*, 1990.

Edward C Tolman. Cognitive maps in rats and men. *Psychological review*, 1948.

Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 2008.

James CR Whittington, Timothy H Muller, Shirley Mark, Guifen Chen, Caswell Barry, Neil Burgess, and Timothy EJ Behrens. The tolman-eichenbaum machine: unifying space and relational memory through generalization in the hippocampal formation. *Cell*, 2020.

Alex Williams, Anthony Degleris, Yixin Wang, and Scott Linderman. Point process models for sequence detection in high-dimensional neural spike trains. *Advances in neural information processing systems*, 2020.

Matthew A Wilson and Bruce L McNaughton. Dynamics of the hippocampal ensemble code for space. *Science*, 1993.

Brian A Wilt, Laurie D Burns, Eric Tatt Wei Ho, Kunal K Ghosh, Eran A Mukamel, and Mark J Schnitzer. Advances in light microscopy for neuroscience. *Annual review of neuroscience*, 2009.

Anqi Wu, Nicholas A Roy, Stephen Keeley, and Jonathan W Pillow. Gaussian process based nonlinear latent structure discovery in multivariate spike train data. *Advances in neural information processing systems*, 2017.

Byron M Yu, Afsheen Afshar, Gopal Santhanam, Stephen Ryu, Krishna V Shenoy, and Maneesh Sahani. Extracting dynamical structure embedded in neural activity. *Advances in neural information processing systems*, 2005.

Byron M Yu, Krishna V Shenoy, and Maneesh Sahani. Expectation propagation for inference in nonlinear dynamical models with poisson observations. In *2006 IEEE Nonlinear Statistical Signal Processing Workshop*. IEEE, 2006.

Byron M Yu, John P Cunningham, Gopal Santhanam, Stephen Ryu, Krishna V Shenoy, and Maneesh Sahani. Gaussian-process factor analysis for low-dimensional single-trial analysis of neural population activity. *Advances in neural information processing systems*, 2008a.

Byron M Yu, John P Cunningham, Krishna V Shenoy, and Maneesh Sahani. Neural decoding of movements: From linear to nonlinear trajectory models. In *Neural Information Processing: 14th International Conference, ICONIP 2007, Kitakyushu, Japan, November 13-16, 2007, Revised Selected Papers, Part I 14*. Springer, 2008b.

Yuan Zhao and Il Memming Park. Variational latent gaussian process for recovering single-trial dynamics from population spike trains. *Neural computation*, 2017.

Ding Zhou and Xue-Xin Wei. Learning identifiable and interpretable latent models of high-dimensional neural activity using pi-vae. *Advances in Neural Information Processing Systems*, 2020.

Petr Znamenskiy and Anthony M Zador. Corticostriatal neurons in auditory cortex drive decisions during auditory discrimination. *Nature*, 2013.

Supplementary Material for "SIMPL: Scalable and hassle-free optimization of neural representations from behaviour"

A Additional Method Details

A.1 Background: Expectation Maximization

Expectation Maximization (EM, Dempster et al. 1977) is a widely used paradigm to perform statistical estimation in latent variable models. The goal of EM is to maximize the *Free Energy*, a lower bound on the log-likelihood $\log p(\mathbf{s}; f)$ of the data, given by (following the notations of Section 2.1):

$$\mathcal{F}(f, q) \coloneqq \mathbb{E}_{q(\mathbf{x})}[\log p(\mathbf{x}, \mathbf{s}\,; f)] - \mathbb{E}_{q(\mathbf{x})}[\log q(\mathbf{x})] \leq \log p(\mathbf{s}; f),$$

where q is some probability distribution on the latent variable $\mathbf{x}$. Importantly, for a given set of intensity functions f, $\mathcal{F}$ is maximized at $q^{\star} = p(\mathbf{x}|\mathbf{s}\,; f)$, i.e. the posterior distribution of the latent variable given the $\mathbf{s}$ and f. Moreover, for a fixed q, the only f-dependent term in $\mathcal{F}$ is $\mathbb{E}_{q(\mathbf{x})}[\log p(\mathbf{x}, \mathbf{s}\,; f)]$. To maximize $\mathcal{F}(f, q)$, EM produces a sequence $(f^{[k]})_{k \geq 0}$ of parameters $f^{[k]}$ by invoking, at each step k and given $f^{[k-1]}$, two well known subroutines:

- **E-step**: Define $q^{[k]} \coloneqq p(\mathbf{x}|\mathbf{s}\,; f^{[k-1]})$; compute $f \longmapsto \mathbb{E}_{q^{[k]}}[\log p(\mathbf{x}, \mathbf{s}\,; f)]$
- **M-step**: Compute $f^{[k]} \coloneqq \arg\max_f \mathcal{F}(f, q^{[k]}) = \arg\max_f \mathbb{E}_{q^{[k]}}[\log p(\mathbf{x}, \mathbf{s}\,; f)]$

with the property that $\log p(\mathbf{s}; f^{[k]}) \geq \log p(\mathbf{s}; f^{[k-1]})$ for all k, thus grounding the use of EM to maximize the likelihood of the data. As the E-step computes specific posterior expectations, a tractable E-step often implies the ability to compute in particular posterior means and variances, the most valuable expectations in the context of decoding the latent variable from behaviour. Thus, in the context of neural data, EM offers a framework to both estimate intensity functions via maximum likelihood, and to decode the variable encoded by the neurons.

A.2 SIMPL as an approximate EM algorithm

Impossibility of Exact EM for Gaussian-Modulated Poisson Processes The E-step of the EM algorithm requires computing a function *defined* as an expectation w.r.t $p(\mathbf{x}|\mathbf{s}\,; f^{[k-1]})$. In the case of Hidden Markov Models, such expectations are intractable to compute in closed form, unless the latent variable $\mathbf{x}$ is discrete, or both the transition and the emission probabilities are Gaussian (with mean and variance depending linearly on $\mathbf{x}$, Rauch et al. 1965). In particular, exact inference in the model described in Section 2.1 is impossible. In order to perform statistical inference for our spike train model, SIMPL runs an approximation of Exact EM, which we detail below.

MLE-backed Approximate E-Step Instead of $q^{[k]} = p(\mathbf{x}|\mathbf{s}\,; f^{[k-1]})$, SIMPL computes an approximation $\widehat{q}^{[k]}$ to $q^{[k]}$, allowing for both statistical estimation and uncertainty-aware trajectory decoding. As a first step towards obtaining $\widehat{q}^{[k]}$, SIMPL first performs Maximum Likelihood Estimation (MLE) on the latent trajectory $\mathbf{x}$. Instead of returning a posterior on $\mathbf{x}$, MLE returns a point estimate of the *true* trajectory that led to the observed spike train $\mathbf{s}$. In particular, MLE does not use the prior knowledge encoded by $p(\mathbf{x})$. The MLE $\widehat{\mathbf{x}}$ of $\mathbf{x}$ given $\mathbf{s}$ is given by:

$$\widehat{\mathbf{x}} = \arg\max_{\mathbf{x}} \log p(\mathbf{s}|\mathbf{x}\,; f^{[k-1]}) = \arg\max_{\mathbf{x}} \sum_{t=1}^{T} \sum_{i=1}^{N} \log p(\mathbf{s}_t^i|\mathbf{x}_t\,; f^{[k-1]})$$

$$\implies \widehat{\mathbf{x}}_t = \arg\max_{\mathbf{x}_t} \sum_{i=1}^{N} \log p(\mathbf{s}_t^i|\mathbf{x}_t\,; f^{[k-1]}).$$

The second equality follows from the conditional independence structure of the HMM. This maximization problem can be solved independently for each t, yielding the formula for $\widehat{\mathbf{x}}_t$ given by the third equality. As a function of $\mathbf{s}$, the MLE $\widehat{\mathbf{x}}$ is itself a random variable. In the many neurons limit, under certain regularity assumption, converges to a Gaussian, a fact known as *asymptotic normality*.

We restate a formal statement of this result in the case of independent, but non identically distributed observations [3] originally established in Bradley & Gart (1962), and reformulated using the notations of the model at hand. For simplicity, we will consider the case where only P distinct intensity functions $f_1, \dots, f_P$ exist, although versions of this result exist without this assumption.

Theorem A.1 (Asymptotic Normality of the MLE)**.** *Let $\mathbf{x}_t \in \mathbb{R}^d$. Let $\mathbf{s}_t^1, \dots, \mathbf{s}_t^N$ be independent random variables with probability densities $p(\mathbf{s}_t^i|\mathbf{x}_t; f_{t(i)})$, where $t(i) \in 1, \dots, P$ is the index of the intensity function $f_{t(i)}$ that generated the spike train $\mathbf{s}_t^i$. For $p \in 1, \dots, P$, denote n_p the number of times the intensity function f_p appeared in the sequence $f_{t(i)}$. Assume that the MLE exists and it is unique. Then, under mild regularity conditions, we have:*

$$\sqrt{N}\left(\widehat{\mathbf{x}}_t - \mathbf{x}_t\right) \xrightarrow[N\to\infty]{\mathrm{d}} \mathcal{N}(0, \mathcal{I}(\mathbf{x}_t)^{-1})$$

where $\mathcal{I}(\mathbf{x}_t) := \sum_{p=1}^{P} \mu_p \mathbb{E}_{p(\mathbf{s}_t;f_p)} \mathrm{Hess}(\log p(\mathbf{s}_t|\mathbf{x}_t; f_p))$ is the Fisher Information matrix and $\overset{\mathrm{d}}{\to}$ means convergence in distribution, and we defined $\mu_p := \lim_{N\to\infty} \frac{n_p}{N}$.

The asymptotic Gaussianity of the MLE in the many neurons limit suggests performing approximate inference in a surrogate Hidden Markov Model, with the same transition probabilities $p(\mathbf{x}_{t+1}|\mathbf{x}_t)$ as the original ones, but where the observations $\mathbf{s}$ are replaced by the previously computed MLE $\widehat{\mathbf{x}}$ of the latent variable. Leveraging Theorem A.1 and the fact that $\mathcal{I}(\mathbf{x}_t) \simeq \mathcal{I}(\widehat{\mathbf{x}}_t)$ SIMPL approximates the emission probabilities $p(\widehat{\mathbf{x}}_t|\mathbf{x}_t)$ by the Gaussian distribution $\mathcal{N}(\mathbf{x}_t, (N\mathcal{I}(\widehat{\mathbf{x}}_t))^{-1})$, treating the covariance matrix as deterministic. The resulting HMM then takes the form of a Linear Gaussian State Space Model, allowing SIMPL to compute the posterior $p(\mathbf{x}_t|\widehat{\mathbf{x}})$ using Kalman Smoothing (Rauch et al., 1965). This posterior is then used as the approximation $\widehat{q}^{[k]}$ to $q^{[k]}$ in SIMPL's E-step. Finally, $\mathcal{F}(f, \widehat{q}^{[k]})$ is approximated by sampling from $\widehat{q}^{[k]}$, and computing the empirical average of $\log p(\mathbf{x}, \mathbf{s}\,; f)$. Importantly, obtaining the MLE estimates $\widehat{\mathbf{x}}_t$ can be obtained in parallel for all t; the only sequential procedure remaining being the Kalman Smoothing step.

Spike Smoothing as an approximate M-Step In the M-step, one maximizes $\mathbb{E}_{\widehat{q}^{[k]}}[\log p(\mathbf{x}, \mathbf{s}\,; f)]$ w.r.t to the intensity functions f. This step is often done by specifying a parametric model for f, and then optimizing the parameters. However, if the true function cannot be accurately represented by the model, the final procedure will suffer from a bias that does not vanish in the large sample limit. While one could use a neural network (whose bias can be made arbitrarily small by increasing the number of neurons), neural networks can be hard to interpret and expensive to train. Instead, SIMPL uses a non-parametric approach that is both training-free and interpretable. To do so, SIMPL samples from its approximate posterior $\tilde{\mathbf{x}} \sim \widehat{q}^{[k]}$, and computes a non-parametric estimate (Hodara et al., 2018) of the intensity functions f_i given by:

$$\widehat{f}_i^{[k]}(x) := \frac{\sum_{t=1}^{T} \mathbf{s}_t^i\, k(x, \tilde{\mathbf{x}}_t)}{\sum_{t=1}^{T} k(x, \tilde{\mathbf{x}}_t)}. \tag{4}$$

Here, $k : \mathbb{R}^d \times \mathbb{R}^d \longmapsto \mathbb{R}_+$ is some kernel function. We propose an explanation of the above formula as the generalization of an M-step: for a fixed $\widehat{q}^{[k]}$, $\mathbb{E}_{p(\mathbf{s})\widehat{q}^{[k]}(\mathbf{x})} \log p(\mathbf{s}, \mathbf{x}\,; f)$ equals (up to a constant) minus the KL divergence between the "data" distribution [4] $p(\mathbf{s})\widehat{q}^{[k]}(\mathbf{x}|\mathbf{s})$ and the model $p(\mathbf{s}, \mathbf{x}; f)$. Thus, an M-step can be understood as minimizing this KL divergence approximately, by replacing the expectation over $p(\mathbf{s})$ by an empirical average over the true data $\mathbf{s}$, an approximation which is asymptotically consistent in the large number of time-steps limit under suitable ergodicity conditions (Billingsley, 1961). SIMPL relaxes this approximation further, replacing the expectation over $\widehat{q}^{[k]}(\mathbf{x}|s)$ by a one-sample estimate of it through $\tilde{\mathbf{x}}$. Moreover, it does not use the KL as a loss function, but instead performs model fitting in a non-parametric manner. Under this procedure, the existing guarantees regarding the EM algorithm do not hold – on the other hand, SIMPL's M-step precisely matches spike smoothing, a standard practice in neuroscience.

[3] The i.i.d case was established in Fisher (1925)

[4] We denote $q^k(x)$ by $q^k(x|s)$ to highlight the dependence between x and s.