

Interrogating theoretical models of neural computation with deep inference

Sean R. Bittner¹, Agostina Palmigiano¹, Alex T. Piet², Chunyu A. Duan³, Carlos D. Brody², Kenneth D. Miller¹, and John P. Cunningham⁴.

¹Department of Neuroscience, Columbia University,

²Princeton Neuroscience Institute,

³Institute of Neuroscience, Chinese Academy of Sciences,

⁴Department of Statistics, Columbia University

¹ 1 Abstract

² A cornerstone of theoretical neuroscience is the circuit model: a system of equations that captures
³ a hypothesized neural mechanism. Such models are valuable when they give rise to an experimen-
⁴ tally observed phenomenon – whether behavioral or in terms of neural activity – and thus can
⁵ offer insights into neural computation. The operation of these circuits, like all models, critically
⁶ depends on the choices of model parameters. Historically, the gold standard has been to analyt-
⁷ ically derive the relationship between model parameters and computational properties. However,
⁸ this enterprise quickly becomes infeasible as biologically realistic constraints are included into the
⁹ model increasing its complexity, often resulting in *ad hoc* approaches to understanding the relation-
¹⁰ ship between model and computation. We bring recent machine learning techniques – the use of
¹¹ deep generative models for probabilistic inference – to bear on this problem, learning distributions
¹² of parameters that produce the specified properties of computation. Importantly, the techniques
¹³ we introduce offer a principled means to understand the implications of model parameter choices
¹⁴ on computational properties of interest. We motivate this methodology with a worked example
¹⁵ analyzing sensitivity in the stomatogastric ganglion. We then use it to go beyond linear theory
¹⁶ of neuron-type input-responsivity in a model of primary visual cortex, gain a mechanistic under-
¹⁷ standing of rapid task switching in superior colliculus models, and attribute error to connectivity
¹⁸ properties in recurrent neural networks solving a simple mathematical task. More generally, this
¹⁹ work suggests a departure from realism vs tractability considerations, towards the use of modern
²⁰ machine learning for sophisticated interrogation of biologically relevant models.

21 2 Introduction

22 The fundamental practice of theoretical neuroscience is to use a mathematical model to understand
23 neural computation, whether that computation enables perception, action, or some intermediate
24 processing [1]. A neural computation is systematized with a set of equations – the model – and
25 these equations are motivated by biophysics, neurophysiology, and other conceptual considerations.
26 The function of this system is governed by the choice of model parameters, which when configured
27 in a particular way, give rise to a measurable signature of a computation. The work of analyzing a
28 model then requires solving the inverse problem: given a computation of interest, how can we reason
29 about these particular parameter configurations? The inverse problem is crucial for reasoning about
30 likely parameter values, uniquenesses and degeneracies, attractor states and phase transitions, and
31 predictions made by the model.

32 Consider the idealized practice: one carefully designs a model and analytically derives how model
33 parameters govern the computation. Seminal examples of this gold standard (which often adopt
34 approaches from statistical physics) include our field’s understanding of memory capacity in asso-
35 ciative neural networks [2], chaos and autocorrelation timescales in random neural networks [3],
36 the paradoxical effect [4], and decision making [5]. Unfortunately, as circuit models include more
37 biological realism, theory via analytical derivation becomes intractable. This creates an unfavor-
38 able tradeoff. On the one hand, one may tractably analyze systems of equations with unrealistic
39 assumptions (for example symmetry or gaussianity), mathematically formalizing how parameters
40 affect computation in a too-simple model. On the other hand, one may choose a more biologically
41 accurate, scientifically relevant model at the cost of *ad hoc* approaches to analysis (such as sim-
42 ply examining simulated activity), potentially resulting in bad inference of parameters and thus
43 erroneous scientific predictions or conclusions.

44 Of course, this same tradeoff has been confronted in many scientific fields characterized by the
45 need to do inference in complex models. In response, the machine learning community has made
46 remarkable progress in recent years, via the use of deep neural networks as a powerful inference
47 engine: a flexible function family that can map observed phenomena (in this case the measurable
48 signal of some computation) back to probability distributions quantifying the likely parameter
49 configurations. One celebrated example of this approach from machine learning, of which we
50 draw key inspiration for this work, is the variational autoencoder [6, 7], which uses a deep neural
51 network to induce an (approximate) posterior distribution on hidden variables in a latent variable

model, given data. Indeed, these tools have been used to great success in neuroscience as well, in particular for interrogating parameters (sometimes treated as hidden states) in models of both cortical population activity [8, 9, 10, 11] and animal behavior [12, 13, 14]. These works have used deep neural networks to expand the expressivity and accuracy of statistical models of neural data [15].

However, these inference tools have not significantly influenced the study of theoretical neuroscience models, for at least three reasons. First, at a practical level, the nonlinearities and dynamics of many theoretical models are such that conventional inference tools typically produce a narrow set of insights into these models. Indeed, only in the last few years has deep learning research advanced to a point of relevance to this class of problem. Second, the object of interest from a theoretical model is not typically data itself, but rather a qualitative phenomenon – inspection of model behavior, or better, a measurable signature of some computation – an *emergent property* of the model. Third, because theoreticians work carefully to construct a model that has biological relevance, such a model as a result often does not fit cleanly into the framing of a statistical model. Technically, because many such models stipulate a noisy system of differential equations that can only be sampled or realized through forward simulation, they lack the explicit likelihood and priors central to the probabilistic modeling toolkit.

To address these three challenges, we developed an inference methodology – ‘emergent property inference’ – which learns a distribution over parameter configurations in a theoretical model. This distribution has two critical properties: (*i*) it is chosen such that draws from the distribution (parameter configurations) correspond to systems of equations that give rise to a specified emergent property (a set of constraints); and (*ii*) it is chosen to have maximum entropy given those constraints, such that we identify all likely parameters and can use the distribution to reason about parametric sensitivity and degeneracies [16]. First, we stipulate a bijective deep neural network that induces a flexible family of probability distributions over model parameterizations with a probability density we can calculate [17, 18, 19]. Second, we quantify the notion of emergent properties as a set of moment constraints on datasets generated by the model. Thus, an emergent property is not a single data realization, but a phenomenon or a feature of the model, which is ultimately the object of interest in theoretical neuroscience. Conditioning on an emergent property requires a variant of deep probabilistic inference methods, which we have previously introduced [20]. Third, because we can not assume the theoretical model has explicit likelihood on data or the emergent property of interest, we use stochastic gradient techniques in the spirit of likelihood free variational inference

[21]. Taken together, emergent property inference (EPI) provides a methodology for inferring parameter configurations consistent with a particular emergent phenomena in theoretical models. We use a classic example of parametric degeneracy in a biological system, the stomatogastric ganglion [22], to motivate and clarify the technical details of EPI.

Equipped with this methodology, we then investigated three models of current importance in theoretical neuroscience. These models were chosen to demonstrate generality through ranges of biological realism (from conductance-based biophysics to recurrent neural networks), neural system function (from pattern generation to abstract cognitive function), and network scale (from four to infinite neurons). First, we use EPI to produce a set of verifiable hypotheses of input-responsivity in a four neuron-type dynamical model of primary visual cortex; we then validate these hypotheses in the model. Second, we demonstrated how the systematic application of EPI to levels of task performance can generate experimentally testable hypotheses regarding connectivity in superior colliculus. Third, we use EPI to uncover the sources of error in a low-rank recurrent neural network executing a simple mathematical task. The novel scientific insights offered by EPI contextualize and clarify the previous studies exploring these models [23, 24, 25, 26], and more generally, these results point to the value of deep inference for the interrogation of biologically relevant models.

We note that, during our preparation and early presentation of this work [27, 28], another work has arisen with broadly similar goals: bringing statistical inference to mechanistic models of neural circuits [29, 30]. We are encouraged by this general problem being recognized by others in the community, and we emphasize that these works offer complementary neuroscientific contributions (different theoretical models of focus) and use different technical methodologies (ours is built on our prior work [20], theirs similarly [31]). These distinct methodologies and scientific investigations emphasize the increased importance and timeliness of both works.

3 Results

3.1 Motivating emergent property inference of theoretical models

Consideration of the typical workflow of theoretical modeling clarifies the need for emergent property inference. First, one designs or chooses an existing model that, it is hypothesized, captures the computation of interest. To ground this process in a well-known example, consider the stomatogastric ganglion (STG) of crustaceans, a small neural circuit which generates multiple rhythmic muscle activation patterns for digestion [32]. Despite full knowledge of STG connectivity and a

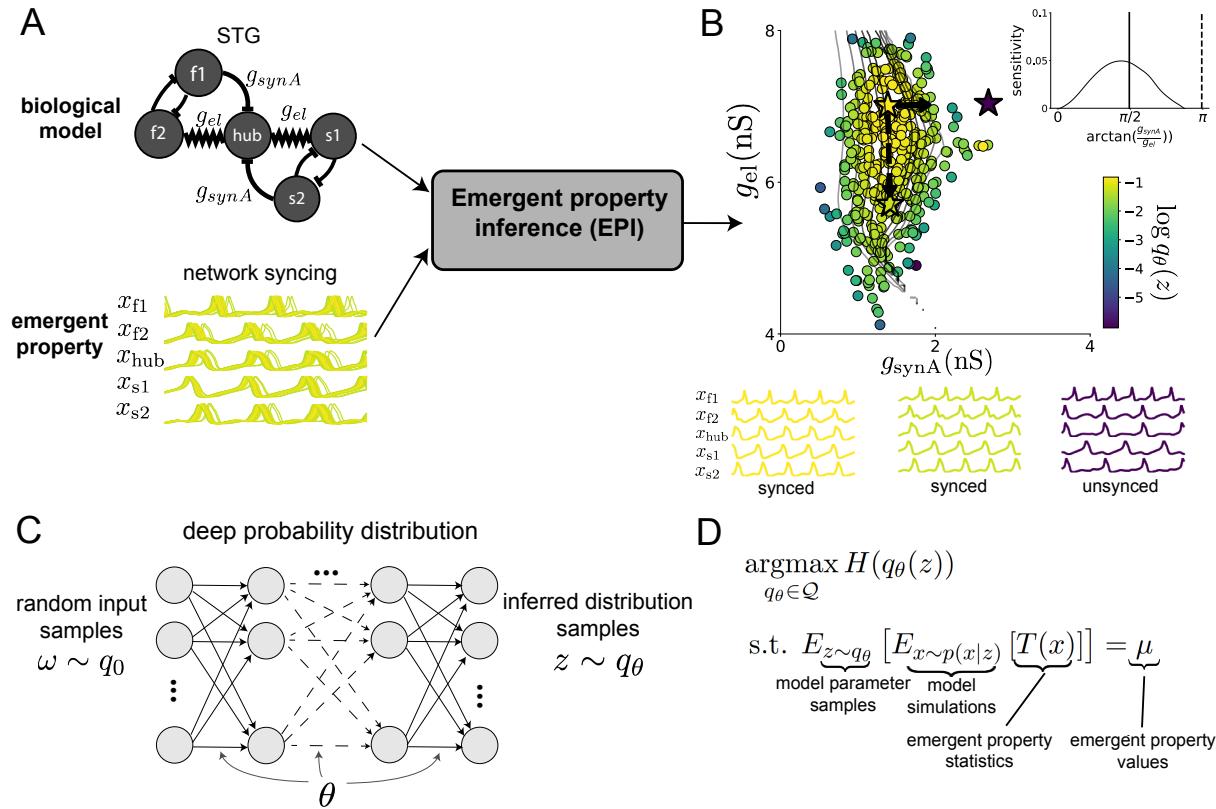


Figure 1: Emergent property inference (EPI) in the stomatogastric ganglion. A. For a choice of model (STG) and emergent property (network syncing), emergent property inference (EPI, gray box) learns a distribution of the model parameters $z = [g_{el}, g_{\text{synA}}]$ producing network syncing. In the STG model, jagged connections indicate electrical coupling having electrical conductance g_{el} . Other connections in the diagram are inhibitory synaptic projections having strength g_{synA} onto the hub neuron, and $g_{\text{synB}} = 5\text{nS}$ for mutual inhibitory connections. Network syncing traces are colored by log probability density of their generating parameters (stars) in the EPI-inferred distribution. B. The EPI distribution of STG model parameters producing network syncing. Samples are colored by log probability density. Distribution contours of emergent property value error are shown at levels of 5×10^{-7} and 1×10^{-6} (dark and light gray). Eigenvectors of the Hessian at the mode of the inferred distribution are indicated as v_1 (solid) and v_2 (dashed) with lengths scaled by the square root of the absolute value of their eigenvalues. Simulated activity is shown for three samples (stars). (Inset) Sensitivity of the system with respect to network syncing along all dimensions of parameter space away from the mode. (see Section B.2.1). C. Deep probability distributions map a latent random variable w through a deep neural network with weights and biases θ to parameters $z = f_\theta(w)$ distributed as $q_\theta(z)$. D. EPI optimization: To learn the EPI distribution $q_\theta(z)$ of model parameters that produce an emergent property, the emergent property statistics $T(x)$ are set in expectation over model parameter samples $z \sim q_\theta(z)$ and model simulations $x \sim p(x | z)$ to emergent property values μ .

114 precise characterization of its rhythmic pattern generation, biophysical models of the STG have
 115 complicated relationships between circuit parameters and neural activity [22, 33]. A model of the
 116 STG [23] is shown schematically in Figure 1A, and note that the behavior of this model will be crit-
 117 ically dependent on its parameterization – the choices of conductance parameters $z = [g_{el}, g_{synA}]$.
 118 Specifically, the two fast neurons (f_1 and f_2) mutually inhibit one another, and oscillate at a faster
 119 frequency than the mutually inhibiting slow neurons (s_1 and s_2). The hub neuron (hub) couples
 120 with either the fast or slow population or both.
 121 Second, once the model is selected, one defines the emergent property, the measurable signal of
 122 scientific interest. To continue our running STG example, one such emergent property is the
 123 phenomenon of *network syncing* – in certain parameter regimes, the frequency of the hub neuron
 124 matches that of the fast and slow populations at an intermediate frequency. This emergent property
 125 is shown in Figure 1A at a frequency of 0.53Hz.
 126 Third, qualitative parameter analysis ensues: since precise mathematical analysis is intractable in
 127 this model, a brute force sweep of parameters is done [23]. Subsequently, a qualitative description
 128 is formulated to describe the different parameter configurations that lead to the emergent property.
 129 In this last step lies the opportunity for a precise quantification of the emergent property as a
 130 statistical feature of the model. Once we have such a methodology, we can infer a probability
 131 distribution over parameter configurations that produce this emergent property.
 132 Before presenting technical details (in the following section), let us understand emergent property
 133 inference schematically: EPI (Fig. 1A gray box) takes, as input, the model and the specified
 134 emergent property, and as its output, produces the parameter distribution shown in Figure 1B.
 135 This distribution – represented for clarity as samples from the distribution – is then a scientifically
 136 meaningful and mathematically tractable object. In the STG model, this distribution can be
 137 specifically queried to reveal the prototypical parameter configuration for network syncing (the
 138 mode; Figure 1B yellow star), and how network syncing decays based on changes away from the
 139 mode. The eigenvectors (of the Hessian of the distribution at the mode) quantitatively formalize
 140 the robustness of network syncing (Fig. 1B v_1 and v_2). Indeed, samples equidistant from the
 141 mode along these EPI-identified dimensions of sensitivity (v_1) and degeneracy (v_2) agree with
 142 error contours (Fig. 1B, contours) and have diminished or preserved network syncing, respectively
 143 (Figure 1B inset and activity traces).

144 3.2 A deep generative modeling approach to emergent property inference

145 Emergent property inference (EPI) systematizes the three-step procedure of the previous section.
 146 First, we consider the model as a coupled set of differential (and potentially stochastic) equations
 147 [23]. In the running STG example, the model activity $x = [x_{f1}, x_{f2}, x_{hub}, x_{s1}, x_{s2}]$ is the membrane
 148 potential for each neuron, which evolves according to the biophysical conductance-based equation:

$$C_m \frac{dx}{dt} = -h(x; z) = -[h_{leak}(x; z) + h_{Ca}(x; z) + h_K(x; z) + h_{hyp}(x; z) + h_{elec}(x; z) + h_{syn}(x; z)] \quad (1)$$

149 where $C_m = 1\text{nF}$, and h_{leak} , h_{Ca} , h_K , h_{hyp} , h_{elec} , and h_{syn} are the leak, calcium, potassium, hyper-
 150 polarization, electrical, and synaptic currents, all of which have their own complicated dependence
 151 on x and $z = [g_{el}, g_{synA}]$ (see Section B.2.1).

152 Second, we define the emergent property, which as above is network syncing: oscillation of the
 153 entire population at an intermediate frequency of our choosing (Figure 1A bottom). Quantifying
 154 this phenomenon is straightforward: we define network syncing to be that each neuron’s spiking
 155 frequency – denoted $\omega_{f1}(x)$, $\omega_{f2}(x)$, etc. – is close to an intermediate frequency of 0.53Hz. Math-
 156 ematically, we achieve this via constraints on the mean and variance of $\omega_\alpha(x)$ for each neuron
 157 $\alpha \in \{f1, f2, hub, s1, s2\}$:

$$\mathbb{E}[T(x)] \triangleq \mathbb{E} \begin{bmatrix} \omega_{f1}(x) \\ \vdots \\ (\omega_{f1}(x) - 0.53)^2 \\ \vdots \end{bmatrix} = \begin{bmatrix} 0.53 \\ \vdots \\ 0.025^2 \\ \vdots \end{bmatrix} \triangleq \mu, \quad (2)$$

158 which completes the quantification of the emergent property.

159 Third, we perform emergent property inference: we find a distribution over parameter configura-
 160 tions z , and insist that samples from this distribution produce the emergent property; in other
 161 words, they obey the constraints introduced in Equation 2. This distribution will be chosen from
 162 a family of probability distributions $\mathcal{Q} = \{q_\theta(z) : \theta \in \Theta\}$, defined by a deep generative distribution
 163 of the normalizing flow class [17, 18, 19] – neural networks which transform a simple distribution
 164 into a suitably complicated distribution (as is needed here). This deep distribution is represented
 165 in Figure 1C (see Section B.1). Then, mathematically, we must solve the following optimization
 166 program:

$$\begin{aligned} & \underset{q_\theta \in \mathcal{Q}}{\operatorname{argmax}} H(q_\theta(z)) \\ & \text{s.t. } \mathbb{E}_{z \sim q_\theta} [\mathbb{E}_{x \sim p(x|z)} [T(x)]] = \mu, \end{aligned} \quad (3)$$

where $T(x), \mu$ are defined as in Equation 2, and $p(x|z)$ is the intractable distribution of data from the model, x , given that model's parameters z (we access samples from this distribution by running the model forward). The purpose of each element in this program is detailed in Figure 1D. Finally, we recognize that many distributions in \mathcal{Q} will respect the emergent property constraints, so we require a normative principle to select amongst them. This principle is captured in Equation 3 by the primal objective H . Here we chose Shannon entropy as a means to find parameter distributions with minimal assumptions beyond some chosen structure [34, 35, 20, 36], but we emphasize that the EPI method is unaffected by this choice (but the results of course will depend on the primal objective chosen).

EPI optimizes the weights and biases θ of the deep neural network (which induces the probability distribution) by iteratively solving Equation 3. The optimization is complete when the sampled models with parameters $z \sim q_\theta$ produce activity consistent with the specified emergent property (Fig. S4). Such convergence is evaluated with a hypothesis test that the mean of each emergent property statistic is not different than its emergent property value (see Section B.1.2). Further validation of EPI is available in the supplementary materials, where we analyze a simpler model for which ground-truth statements can be made (Section B.1.1). In relation to broader methodology, inspection of the EPI objective reveals a natural relationship to posterior inference. Specifically, EPI executes variational inference in an exponential family model, the sufficient statistics and mean parameter of which are defined by the emergent property statistics and values, respectively (see Section B.1.4). Equipped with this method, we now prove out the value of EPI by using it to investigate and produce novel insights about three prominent models in neuroscience.

3.3 Comprehensive input-responsivity in a nonlinear sensory system

Dynamical models of excitatory (E) and inhibitory (I) populations with supralinear input-output function have succeeded in explaining a host of experimentally documented phenomena. In a regime characterized by inhibitory stabilization of strong recurrent excitation, these models give rise to paradoxical responses [4], selective amplification [37], surround suppression [38] and normalization [39]. Despite their strong predictive power, E-I circuit models rely on the assumption that inhibition can be studied as an indivisible unit. However, experimental evidence shows that inhibition is composed of distinct elements – parvalbumin (P), somatostatin (S), VIP (V) – composing 80% of GABAergic interneurons in V1 [40, 41, 42], and that these inhibitory cell types follow specific connectivity patterns (Fig. 2A) [43]. Recent theoretical advances [24, 44, 45], have only started

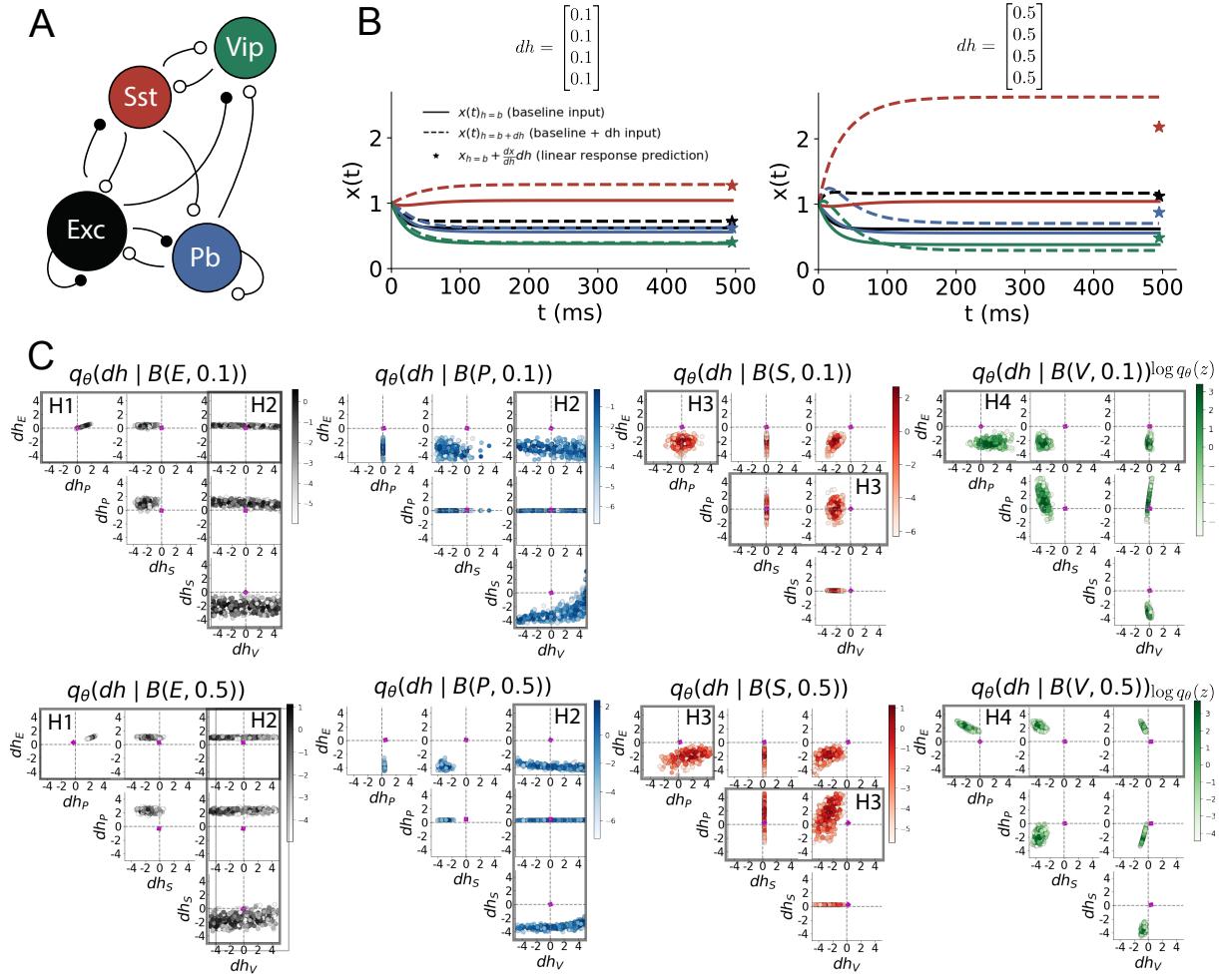


Figure 2: Hypothesis generation through EPI in a V1 model. A. Four-population model of primary visual cortex with excitatory (black), parvalbumin (blue), somatostatin (red), and VIP (green) neurons. Some neuron-types largely do not form synaptic projections to others (excitatory and inhibitory projections filled and unfilled, respectively). B. V1 model simulations for input (solid) $h = b$ and (dashed) $h = b + dh$. Stars indicate the linear response prediction. C. EPI distributions on differential input dh conditioned on differential response $\mathcal{B}(\alpha, y)$. Supporting evidence for the four generated hypotheses are indicated by gray boxes with labels H1, H2, H3, and H4. The linear prediction from two standard deviations away from y (from negative to positive) is overlaid in magenta (very small, near origin).

198 to address the consequences of this multiplicity in the dynamics of V1, strongly relying on linear
 199 theoretical tools. Here, we go beyond linear theory by systematically generating and evaluating hy-
 200 potheses of circuit model function using EPI distributions of neuron-type inputs producing various
 201 neuron-type population responses.

202 Specifically, we consider a four-dimensional circuit model with dynamical state given by the firing
 203 rate x of each neuron-type population $x = [x_E, x_P, x_S, x_V]^\top$. Given a time constant of $\tau = 20$ ms
 204 and a power $n = 2$, the dynamics are driven by the rectified and exponentiated sum of recurrent
 205 (Wx) and external h inputs:

$$\tau \frac{dx}{dt} = -x + [Wx + h]_+^n. \quad (4)$$

206 The effective connectivity weights W were obtained from experimental recordings of publicly avail-
 207 able datasets of mouse V1 [46, 47] (see Section B.2.2). The input $h = b + dh$ is comprised of a
 208 baseline input $b = [b_E, b_P, b_S, b_V]^\top$ and a differential input $dh = [dh_E, dh_P, dh_S, dh_V]^\top$ to each
 209 neuron-type population. Throughout subsequent analyses, the baseline input is $b = [1, 1, 1, 1]^\top$.

210 With this model, we are interested in the differential responses of each neuron-type population to
 211 changes in input dh . Initially, we studied the linearized response of the system to input $\frac{dx_{ss}}{dh}$ at the
 212 steady state response x_{ss} , i.e. a fixed point. All analyses of this model consider the steady state
 213 response, so we drop the notation ss from here on. While this linearization accurately predicts
 214 differential responses $dx = [dx_E, dx_P, dx_S, dx_V]$ for small differential inputs to each population
 215 $dh = [0.1, 0.1, 0.1, 0.1]$ (Fig 2B left), the linearization is a poor predictor in this nonlinear model
 216 more generally (Fig. 2B right). Currently available approaches to deriving the steady state response
 217 of the system are limited.

218 To get a more comprehensive picture of the input-responsivity of each neuron-type beyond linear
 219 theory, we used EPI to learn a distribution of the differential inputs to each population dh that
 220 produce an increase of y in the rate of each neuron-type population $\alpha \in \{E, P, S, V\}$. We want
 221 to know the differential inputs dh that result in a differential steady state dx_α (the change in x_α
 222 when receiving input $h = b + dh$ with respect to the baseline $h = b$) of value y with some small,
 223 arbitrarily chosen amount of variance 0.01^2 . These statements amount to the emergent property

$$\mathcal{B}(\alpha, y) \triangleq \mathbb{E} \begin{bmatrix} dx_\alpha \\ (dx_\alpha - y)^2 \end{bmatrix} = \begin{bmatrix} y \\ 0.01^2 \end{bmatrix} \quad (5)$$

224 We maintain the notation $\mathcal{B}(\cdot)$ throughout the rest of the study as short hand for emergent property,

which represents a different signature of computation in each application.

Using EPI, we inferred the distribution of dh shown in Figure 2C producing $\mathcal{B}(\alpha, y)$. Columns correspond to inferred distributions of excitatory ($\alpha = E$, red), parvalbumin ($\alpha = P$, blue), somatostatin ($\alpha = S$, red) and VIP ($\alpha = V$, green) neuron-type response increases, while each row corresponds to increase amounts of $y \in \{0.1, 0.5\}$. For each pair of parameters, we show the two-dimensional marginal distribution of samples colored by $\log q_\theta(dh | \mathcal{B}(\alpha, y))$. The inferred distributions immediately suggest four hypotheses:

232

- 233 H1: as is intuitive, each neuron-type's firing rate should be sensitive to that neuron-type's
234 direct input (e.g. Fig. 2C H1 gray boxes indicate low variance in dh_E when $\alpha = E$. Same
235 observation in all inferred distributions);
- 236 H2: the E- and P-populations should be largely unaffected by input to the V-population (Fig.
237 2C H2 gray boxes indicate high variance in dh_V when $\alpha \in \{E, P\}$);
- 238 H3: the S-population should be largely unaffected by input to the P-population (Fig. 2C H3
239 gray boxes indicate high variance in dh_P when $\alpha = S$);
- 240 H4: there should be a nonmonotonic response of the V-population with input to the E-
241 population (Fig. 2C H4 gray boxes indicate that negative dh_E should result in small dx_V ,
242 but positive dh_E should elicit a larger dx_V);

243 We evaluate these hypotheses by taking steps in individual neuron-type input δh_α away from the
244 modes of the inferred distributions at $y = 0.1$

$$dh^* = z^* = \underset{z}{\operatorname{argmax}} \log q_\theta(z | \mathcal{B}(\alpha, 0.1)). \quad (6)$$

245 Here δx_α is the change in steady state response to the system with input $h = b + dh^* + \delta h_\alpha \hat{u}_\alpha$
246 compared to $h = b + dh^*$, where \hat{u}_α is a unit vector in the dimension of α . The EPI-generated
247 hypotheses are confirmed (for details, see Section B.2.2):

- 248 H1: the neuron-type responses are sensitive to their direct inputs (Fig. 3A black, 3B blue,
249 3C red, 3D green);
- 250 H2: the E- and P-populations are not affected by δh_V (Fig. 3A green, 3B green);
- 251 H3: the S-population is not affected by δh_P (Fig. 3C blue);
- 252 H4: the V-population exhibits a nonmonotonic response to δh_E (Fig. 3D black), and is in
253 fact the only population to do so (Fig. 3A-C black).

254 These hypotheses were in stark contrast to what was available to us via traditional analytical

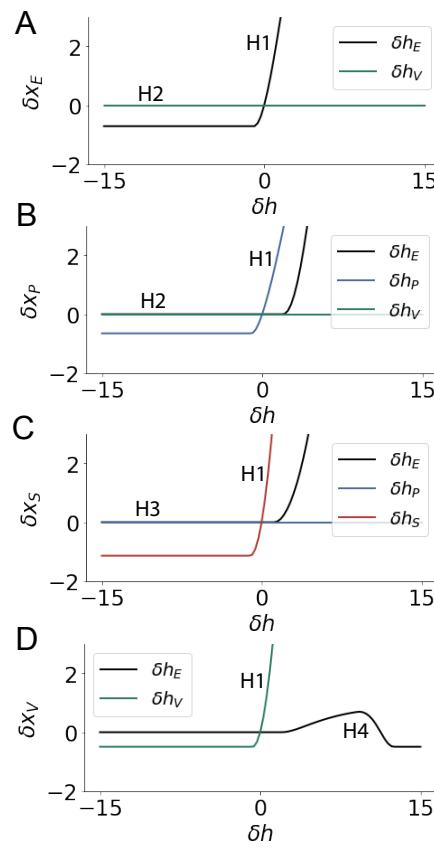


Figure 3: Confirming EPI generated hypotheses in V1. A. Differential responses δx_E by the E-population to changes in individual input $\delta h_\alpha \hat{u}_\alpha$ away from the mode of the EPI distribution dh^* . B-D Same plots for the P-, S-, and V-populations. Labels H1, H2, H3, and H4 indicate which curves confirm which hypotheses.

255 linear prediction (Fig. 2C, magenta, see Section B.2.2). To this point, we have shown the utility of
 256 EPI on relatively low-level emergent properties like network syncing and differential neuron-type
 257 population responses. In the remainder of the study, we focus on using EPI to understand models
 258 of more abstract cognitive function.

259 3.4 Identifying neural mechanisms of flexible task switching

260 In a rapid task switching experiment [48], rats were explicitly cued on each trial to either orient
 261 towards a visual stimulus in the Pro (P) task or orient away from a visual stimulus in the Anti
 262 (A) task (Fig. 4a). Neural recordings in the midbrain superior colliculus (SC) exhibited two
 263 populations of neurons that simultaneously represented both task context (Pro or Anti) and motor
 264 response (contralateral or ipsilateral to the recorded side): the Pro/Contra and Anti/Ipsi neurons
 265 [25]. Duan et al. proposed a model of SC that, like the V1 model analyzed in the previous section, is
 266 a four-population dynamical system. We analyzed this model, where the neuron-type populations
 267 are functionally-defined as the Pro- and Anti-populations in each hemisphere (left (L) and right
 268 (R)), their connectivity is parameterized geometrically (Fig. 4B). The input-output function of this

269 model is chosen such that the population responses $x = [x_{LP} \ x_{LA} \ x_{RP} \ x_{RA}]^\top$ are bounded
 270 from 0 to 1 giving rise to high (1) or low (0) responses at the end of the trial:

$$x_\alpha = \left(\frac{1}{2} \tanh \left(\frac{u_\alpha - \epsilon}{\zeta} \right) + \frac{1}{2} \right) \quad (7)$$

271 where $\epsilon = 0.05$ and $\zeta = 0.5$. The dynamics evolve with timescale $\tau = 0.09$ via an internal variable
 272 u governed by connectivity weights W

$$\tau \frac{du}{dt} = -u + Wx + h + \sigma dB \quad (8)$$

273 with gaussian noise of variance $\sigma^2 = 1$. The input h is comprised of a cue-dependent input to the
 274 Pro or Anti populations, a stimulus orientation input to either the Left or Right populations, and
 275 a choice-period input to the entire network (see Section B.2.3). Here, we use EPI to determine the
 276 changes in network connectivity $z = [sW_P \ sW_A \ vW_{PA} \ vW_{AP} \ dW_{PA} \ dW_{AP} \ hW_P \ hW_A]$
 277 resulting in greater levels of rapid task switching accuracy.

278 To quantify the emergent property of rapid task switching at various levels of accuracy, we consid-
 279 ered the requirements of this model in this behavioral paradigm. At the end of successful trials,
 280 the response of the Pro population in the hemisphere of the correct choice must have a value near
 281 1, while the Pro population in the opposite hemisphere must have a value near 0. Constraining a
 282 population response $x_\alpha \in [0, 1]$ to be either 0 or 1 can be achieved by requiring that it has Bernoulli
 283 variance (see Section B.2.3). Thus, we can formulate rapid task switching at a level of accuracy
 284 $p \in [0, 1]$ in both tasks in terms of the average steady response of the Pro population \hat{p} of the
 285 correct choice, the error in Bernoulli variance of that Pro neuron σ_{err}^2 , and the average difference
 286 in Pro neuron responses d in both Pro and Anti trials:

$$\mathcal{B}(p) \triangleq \mathbb{E} \begin{bmatrix} \hat{p}_P \\ \hat{p}_A \\ (\hat{p}_P - p)^2 \\ (\hat{p}_A - p)^2 \\ \sigma_{P,err}^2 \\ \sigma_{A,err}^2 \\ d_P \\ d_A \end{bmatrix} = \begin{bmatrix} p \\ p \\ 0.15^2 \\ 0.15^2 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix} \quad (9)$$

287 Thus, $\mathcal{B}(p)$ denotes Bernoulli, winner-take-all responses between Pro neurons in a model executing
 288 rapid task switching near accuracy level p .

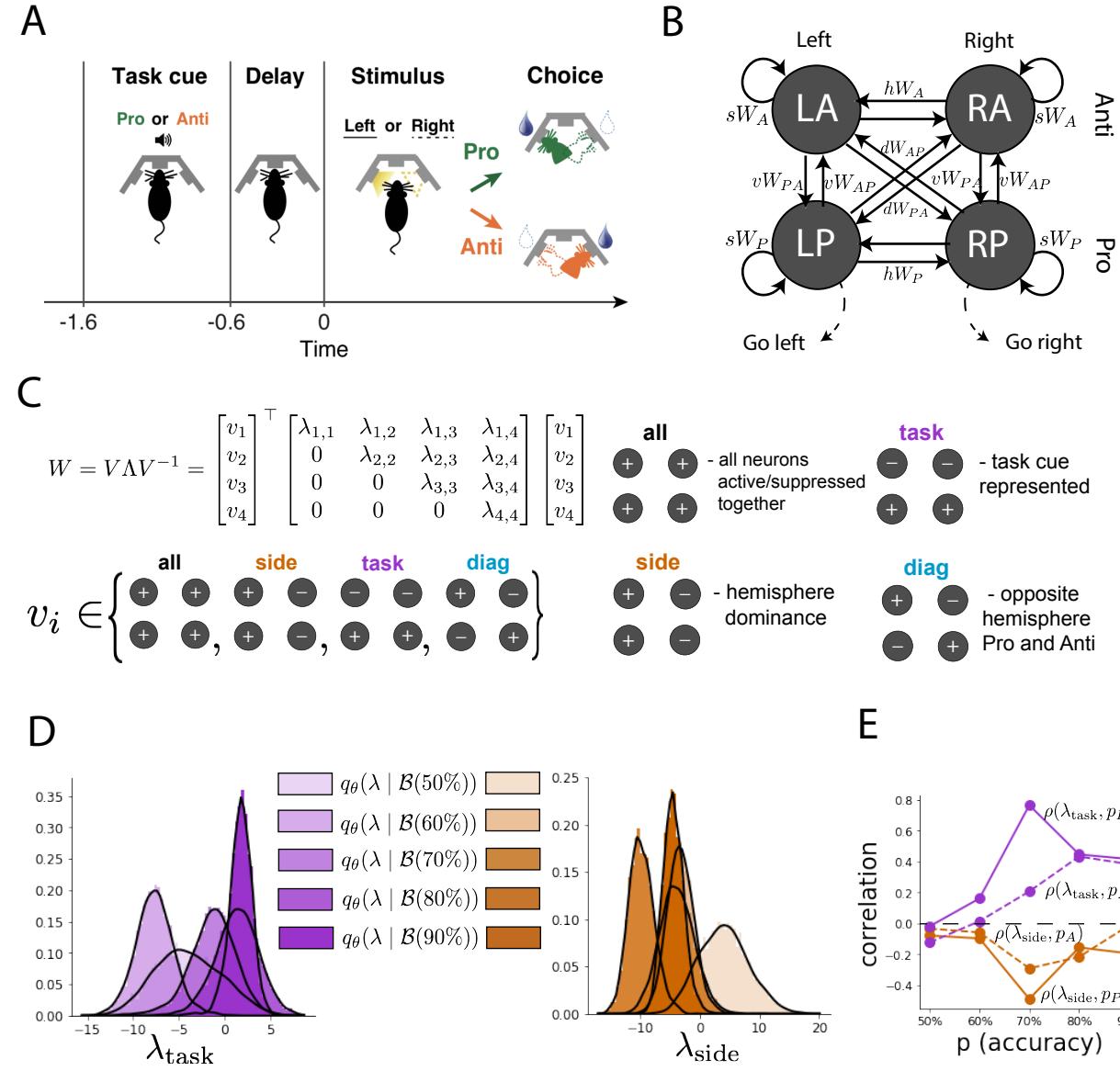


Figure 4: EPI reveals changes in SC [25] connectivity that control task accuracy. A. Rapid task switching behavioral paradigm (see text). B. Model of superior colliculus (SC). Neurons: LP - left pro, RP - right pro, LA - left anti, RA - right anti. Parameters: sW - self, hW - horizontal, vW - vertical, dW - diagonal weights. Subscripts P and A of connectivity weights indicate Pro or Anti populations, and e.g. vW_{PA} is a vertical weight from an Anti to a Pro population. C. The Schur decomposition of the weight matrix $W = V \Lambda V^{-1}$ is a unique decomposition with orthogonal V and upper triangular Λ . Schur modes: v_{all} , v_{task} , v_{side} , and v_{diag} . D. The marginal EPI distributions of the Schur eigenvalues at each level of task accuracy. E. The correlation of Schur eigenvalue with task performance in each learned EPI distribution.

We used EPI to learn distributions of the SC weight matrix parameters z conditioned on various levels of rapid task switching accuracy $\mathcal{B}(p)$ for $p \in \{50\%, 60\%, 70\%, 80\%, 90\%\}$. To make sense of these inferred distributions, we followed the approach of Duan et al. by decomposing the connectivity matrix $W = V\Lambda V^{-1}$ in such a way (the Schur decomposition) that the basis vectors v_i are the same for all W (Fig. 4C). These basis vectors have intuitive roles in processing for this task, and are accordingly named the *all* mode - all neurons co-fluctuate, *side* mode - one side dominates the other, *task* mode - the Pro or Anti populations dominate the other, and *diag* mode - Pro- and Anti-populations of opposite hemispheres dominate the opposite pair. The corresponding eigenvalues (e.g. λ_{task} , which change according to W) indicate the degree to which activity along that mode is increased or decreased by W .

We found that for greater task accuracies, the task mode eigenvalue increases, indicating the importance of W to the task representation (Fig. 4D, purple). Specifically,

$\mathbb{E}_{q_\theta(z|\mathcal{B}(70\%)} [\lambda_{\text{task}}(z)] < \mathbb{E}_{q_\theta(z|\mathcal{B}(80\%)} [\lambda_{\text{task}}(z)]$ (p-value= 3.53×10^{-18} Mann-Whitney test with 50 estimates using 100 samples $z \sim q_\theta(z | \mathcal{B})$), $\mathbb{E}_{q_\theta(z|\mathcal{B}(70\%)} [\lambda_{\text{task}}(z)] < \mathbb{E}_{q_\theta(z|\mathcal{B}(80\%)} [\lambda_{\text{task}}(z)]$ (p-value= 3.53×10^{-18}), and $\mathbb{E}_{q_\theta(z|\mathcal{B}(80\%)} [\lambda_{\text{task}}(z)] < \mathbb{E}_{q_\theta(z|\mathcal{B}(90\%)} [\lambda_{\text{task}}(z)]$ (p-value= 5.23×10^{-14}). Stepping from random chance (50%) networks to marginally task-performing (60%) networks, there is a marked decrease of the side mode eigenvalues $\mathbb{E}_{q_\theta(z|\mathcal{B}(60\%)} [\lambda_{\text{side}}(z)] < \mathbb{E}_{q_\theta(z|\mathcal{B}(50\%)} [\lambda_{\text{side}}(z)]$ (p-value= 3.53×10^{-18}) (Fig. 4D, orange). Such side mode suppression relative to 50% remains in the models achieving greater accuracy, revealing its importance towards task performance (p-value= 3.53×10^{-18} for all accuracies). There were no interesting trends with task accuracy in the all or diag mode (hence not shown in Fig. 4). Importantly, we can conclude from our methodology that side mode suppression in W allows rapid task switching, and that greater task-mode representations in W increase accuracy. These hypotheses are confirmed by forward simulation of the SC model (Fig. 4E, see Section B.2.3) suggesting experimentally testable predictions: increase in rapid task switching performance should be correlated with changes in effective connectivity resulting in an increase in task mode and decrease in side mode eigenvalues.

3.5 Linking RNN connectivity to error

So far, each model we have studied was designed from fundamental biophysical principles, genetically- or functionally-defined neuron types. At a more abstract level of modeling, recurrent neural networks (RNNs) are high-dimensional dynamical models of computation that are becoming increasingly popular in neuroscience research [49]. In theoretical neuroscience, RNN dynamics usually

320 follow the equation

$$\frac{dx}{dt} = -x + W\phi(x) + h, \quad (10)$$

321 where x is the network activity, W is the network connectivity, $\phi(\cdot) = \tanh(\cdot)$, and h is the input to
 322 the system. Such RNNs are trained to do a task from a systems neuroscience experiment, and then
 323 the unit activations of the trained RNN are compared to recorded neural activity. Fully-connected
 324 RNNs with tens of thousands of parameters are challenging to characterize [50], especially making
 325 statistical inferences about their parameterization. Alternatively, we considered a rank-1, N -neuron
 326 RNN with connectivity

$$W = g\chi + \frac{1}{N}mn^\top, \quad (11)$$

327 where $\chi_{i,j} \sim \mathcal{N}(0, \frac{1}{N})$, g is the random strength, and the entries of m and n are drawn from Gaussian
 328 distributions $m_i \sim \mathcal{N}(M_m, 1)$ and $n_i \sim \mathcal{N}(M_n, 1)$. We used EPI to infer the parameterizations of
 329 rank-1 RNNs solving an example task, enabling discovery of properties of connectivity that result
 330 in different types of error in the computation.

331 The task we consider is Gaussian posterior conditioning: calculate the parameters of a posterior
 332 distribution induced by a prior $p(\mu_y) = \mathcal{N}(\mu_0 = 4, \sigma_0^2 = 1)$ and a likelihood $p(y|\mu_y) = \mathcal{N}(\mu_y, \sigma_y^2 =$
 333 $1)$, given a single observation y . Conjugacy offers the result analytically; $p(\mu_y|y) = \mathcal{N}(\mu_{post}, \sigma_{post}^2)$,
 334 where:

$$\mu_{post} = \frac{\frac{\mu_0}{\sigma_0^2} + \frac{y}{\sigma_y^2}}{\frac{1}{\sigma_0^2} + \frac{1}{\sigma_y^2}} \quad \sigma_{post}^2 = \frac{1}{\frac{1}{\sigma_0^2} + \frac{1}{\sigma_y^2}}. \quad (12)$$

335 The RNN is trained to solve this task by producing readout activity that is on average the posterior
 336 mean μ_{post} , and activity whose variability is the posterior variance σ_{post}^2 (Fig. 5A, a setup inspired
 337 by [51]). To solve this Gaussian posterior conditioning task, the RNN response to a constant input
 338 $h = yw + (n - M_n)$ must equal the posterior mean along readout vector r , where

$$\kappa_r = \frac{1}{N} \sum_{j=1}^N r_j \phi(x_j) \quad (13)$$

339 Additionally, the amount of chaotic variance Δ_T must equal the posterior variance. Theory for
 340 low-rank RNNs allows us to express κ_r and Δ_T in terms of each other through a solvable system
 341 of nonlinear equations (see Section B.2.4) [26]. This allows us to mathematically formalize the
 342 execution of this task into an emergent property, where the emergent property statistics of the
 343 RNN activity are κ_r and Δ_T and the emergent property values are the ground truth posterior

³⁴⁴ mean μ_{post} and variance σ_{post}^2 :

$$\mathbb{E} \begin{bmatrix} \kappa_r \\ \Delta_T \\ (\kappa_r - \mu_{\text{post}})^2 \\ (\Delta_T^2 - \sigma_{\text{post}}^2)^2 \end{bmatrix} = \begin{bmatrix} \mu_{\text{post}} \\ \sigma_{\text{post}}^2 \\ 0.1 \\ 0.1 \end{bmatrix} \quad (14)$$

³⁴⁵ We specify a substantial amount of variance in these emergent property statistics, so that the
³⁴⁶ inferred distribution results in RNNs with a variety errors in their solutions to the gaussian posterior
³⁴⁷ conditioning problem.

³⁴⁸ We used EPI to learn distributions of RNN connectivity properties $z = [g \ M_m \ M_n]$ executing
³⁴⁹ Gaussian posterior conditioning given an input of $y = 2$, where the true posterior is $\mu_{\text{post}} = 3$ and
³⁵⁰ $\sigma_{\text{post}} = 0.5$ (see Section B.2.4) (Fig. 5B). We examined the nature of the over- and under-estimation
³⁵¹ of the posterior means (Fig. 5B, left) and variances (Fig. 5B, right) in the inferred distributions.
³⁵² There is rough symmetry in the M_m - M_n plane, suggesting a degeneracy in the product of M_m
³⁵³ and M_n (Fig. 5B). The product of M_m and M_n strongly determines the posterior mean (Fig. 5B,
³⁵⁴ left), and the random strength g is the most influential variable on the chaotic variance (Fig. 5B,
³⁵⁵ right). Neither of these observations were obvious from what mathematical analysis is available in
³⁵⁶ networks of this type (see Section B.2.4). While the relationship of the random strength to chaotic
³⁵⁷ variance (and resultingly posterior variance in this problem) is well-known [3], the distribution
³⁵⁸ admits a hypothesis: the estimation of the posterior mean by the RNN increases with the product
³⁵⁹ of M_m and M_n .

³⁶⁰ We tested this prediction by taking parameters z_1 and z_2 as representative samples from the positive
³⁶¹ and negative M_m - M_n quadrants, respectively. Instead of using the theoretical predictions shown
³⁶² in Figure 5B, we simulated finite-size realizations of these networks with 2,000 neurons (e.g. Fig.
³⁶³ 5C). We perturbed these parameter choices by the product $M_m M_n$ clarifying that the posterior
³⁶⁴ mean can be directly controlled in this way (Fig. 5D). Thus, EPI confers a clear picture of error in
³⁶⁵ this computation: the product of the low rank vector means M_m and M_n modulates the estimated
³⁶⁶ posterior mean while the random strength g modulates the estimated posterior variance. This
³⁶⁷ novel procedure of inference on reduced parameterizations of RNNs conditioned on the emergent
³⁶⁸ property of task execution is generalizable to other settings modeled in [26] like noisy integration
³⁶⁹ and context-dependent decision making (Fig. S5).

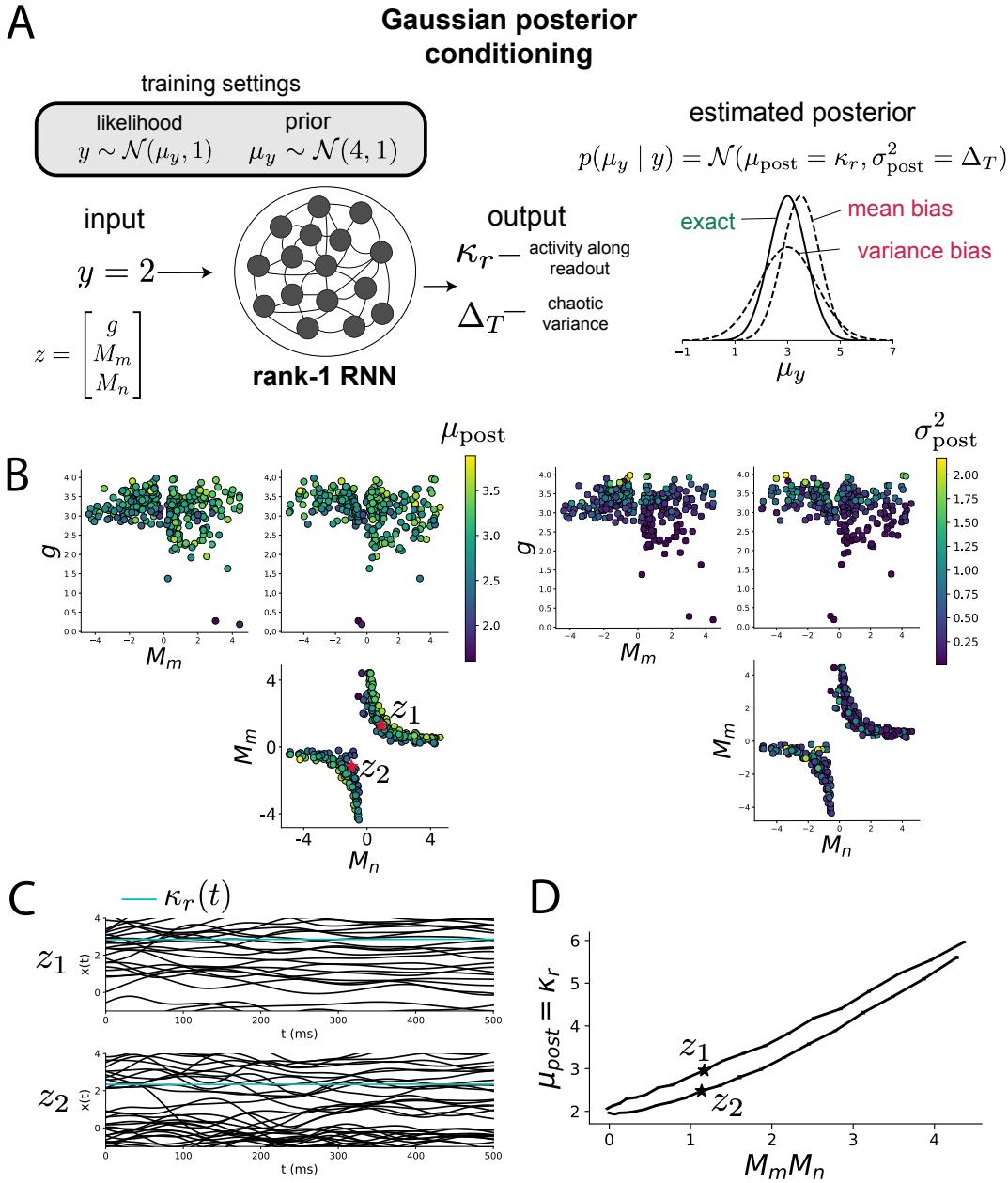


Figure 5: Sources of error in an RNN solving a simple task. A. (left) A rank-1 RNN executing a Gaussian posterior conditioning computation on μ_y . (right) Error in this computation can come from over- or under-estimating the posterior mean or variance. B. EPI distribution of rank-1 RNNs executing Gaussian posterior conditioning. Samples are colored by (left) posterior mean $\mu_{\text{post}} = \kappa_r$ and (right) posterior variance $\sigma_{\text{post}}^2 = \Delta_T$. C. Finite-size network simulations of 2,000 neurons with parameters z_1 and z_2 sampled from the inferred distribution. Activity along readout κ_r (cyan) is stable despite chaotic fluctuations. D. The posterior mean computed by RNNs parameterized by z_1 and z_2 pertrubed in the dimension of the product of M_m and M_n . Means and standard errors are shown across 10 realizations of 2,000-neuron networks.

370 4 Discussion**371 4.1 EPI is a general tool for theoretical neuroscience**

372 Biologically realistic models of neural circuits are comprised of complex nonlinear differential equa-
373 tions, making traditional theoretical analysis and statistical inference intractable. In contrast, EPI
374 is capable of learning distributions of parameters in such models producing measurable signatures
375 of computation. We have demonstrated its utility on biological models (STG), intermediate-level
376 models of interacting genetically- and functionally-defined neuron-types (V1, SC), and the most
377 abstract of models (RNNs). We are able to condition both deterministic and stochastic models on
378 low-level emergent properties like spiking frequency of membrane potentials, as well as high-level
379 cognitive function like posterior conditioning. Technically, EPI is tractable when the emergent
380 property statistics are continuously differentiable with respect to the model parameters, which is
381 very often the case; this emphasizes the general applicability of EPI.

382 In this study, we have focused on applying EPI to low dimensional parameter spaces of models
383 with low dimensional dynamical states. These choices were made to present the reader with a
384 series of interpretable conclusions, which is more challenging in high dimensional spaces. In fact,
385 EPI should scale reasonably to high dimensional parameter spaces, as the underlying technology has
386 produced state-of-the-art performance on high-dimensional tasks such as texture generation [20]. Of
387 course, increasing the dimensionality of the dynamical state of the model makes optimization more
388 expensive, and there is a practical limit there as with any machine learning approach. Although,
389 theoretical approaches (e.g. [26]) can be used to reason about the wholistic activity of such high
390 dimensional systems by introducing some degree of additional structure into the model.

391 There are additional technical considerations when assessing the suitability of EPI for a particu-
392 lar modeling question. First and foremost, as in any optimization problem, the defined emergent
393 property should always be appropriately conditioned (constraints should not have wildly different
394 units). Furthermore, if the program is underconstrained (not enough constraints), the distribution
395 grows (in entropy) unstably unless mapped to a finite support. If overconstrained, there is no pa-
396 rameter set producing the emergent property, and EPI optimization will fail (appropriately). Next,
397 one should consider the computational cost of the gradient calculations. In the best circumstance,
398 there is a simple, closed form expression (e.g. Section B.1.1) for the emergent property statistic
399 given the model parameters. On the other end of the spectrum, many forward simulation iterations
400 may be required before a high quality measurement of the emergent property statistic is available

401 (e.g. Section B.2.1). In such cases, optimization will be expensive.

402 **4.2 Novel hypotheses from EPI**

403 In neuroscience, machine learning has primarily been used to revealed structure in large-scale neural
404 datasets [52, 53, 54, 55, 56, 57] (see review, [15]). Such careful inference procedures are developed
405 for these statistical models allowing precise, quantitative reasoning, which clarifies the way data
406 informs knowledge of the model parameters. However, these inferable statistical models lack re-
407 semblance to the underlying biology, making it unclear how to go from the structure revealed by
408 these methods, to the neural mechanisms giving rise to it. In contrast, theoretical neuroscience has
409 focused on careful mechanistic modeling and the production of emergent properties of computation.
410 The careful steps of 1.) model design and 2.) emergent property definition, are followed by 3.)
411 practical inference methods resulting in an opaque characterization of the way model parameters
412 govern computation. In this work, we replaced this opaque procedure of parameter identification
413 in theoretical neuroscience with emergent property inference, opening the door to careful inference
414 in careful models of neural computation.

415 Biologically realistic models of neural circuits often prove formidable to analyze. For example,
416 consider the fact that we do not fully understand the (only) four-dimensional models of V1 [24]
417 and SC [25]. Because analytical approaches to studying nonlinear dynamical systems become
418 increasingly complicated when stepping from two-dimensional to three- or four-dimensional systems
419 in the absence of restrictive simplifying assumptions [58], it is unsurprising that these models pose a
420 challenge. In Section 3.3, we showed that EPI was far more informative about neuron-type input-
421 responsivity than the predictions afforded through the available linear analytical methods. By
422 flexibly conditioning this V1 model on different emergent properties, we performed an exploratory
423 analysis of a *model* rather than a dataset, which generated a set of testable hypotheses, which
424 were proved out. Of course, exploratory analyses can be directed towards formulating hypotheses
425 of a specific form. For example, when interested in model parameter changes with behavioral
426 performance, one can use EPI to condition on various levels of task accuracy as we did in Section
427 3.4. This analysis identified experimentally testable predictions (proved out *in-silico*) of patterns
428 of effective connectivity in SC that should be correlated with increased performance.

429 In our final analysis, we presented a novel procedure for doing statistical inference on interpretable
430 parameterizations of RNNs executing simple tasks. Specifically, we analyzed RNNs solving a pos-
431 terior conditioning problem in the spirit of [51]. This methodology relies on recently extended

432 theory of responses in random neural networks with minimal structure [26]. While we focused on
433 rank-1 RNNs, which were sufficient for solving this task, we can more generally use this approach
434 to analyze rank-2 and greater RNNs. The ability to apply the probabilistic model selection toolkit
435 to such black box models should prove invaluable as their use in neuroscience increases.

436 **References**

- 437 [1] Larry F Abbott. Theoretical neuroscience rising. *Neuron*, 60(3):489–495, 2008.
- 438 [2] John J Hopfield. Neural networks and physical systems with emergent collective computational
439 abilities. *Proceedings of the national academy of sciences*, 79(8):2554–2558, 1982.
- 440 [3] Haim Sompolinsky, Andrea Crisanti, and Hans-Jurgen Sommers. Chaos in random neural
441 networks. *Physical review letters*, 61(3):259, 1988.
- 442 [4] Misha V Tsodyks, William E Skaggs, Terrence J Sejnowski, and Bruce L McNaughton. Para-
443 doxical effects of external modulation of inhibitory interneurons. *Journal of neuroscience*,
444 17(11):4382–4388, 1997.
- 445 [5] Kong-Fatt Wong and Xiao-Jing Wang. A recurrent network mechanism of time integration in
446 perceptual decisions. *Journal of Neuroscience*, 26(4):1314–1328, 2006.
- 447 [6] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *International Confer-
448 ence on Learning Representations*, 2014.
- 449 [7] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation
450 and variational inference in deep latent gaussian models. *International Conference on Machine
451 Learning*, 2014.
- 452 [8] Yuanjun Gao, Evan W Archer, Liam Paninski, and John P Cunningham. Linear dynamical
453 neural population models through nonlinear embeddings. In *Advances in neural information
454 processing systems*, pages 163–171, 2016.
- 455 [9] Yuan Zhao and Il Memming Park. Recursive variational bayesian dual estimation for nonlinear
456 dynamics and non-gaussian observations. *stat*, 1050:27, 2017.
- 457 [10] Gabriel Barello, Adam Charles, and Jonathan Pillow. Sparse-coding variational auto-encoders.
458 *bioRxiv*, page 399246, 2018.

- 459 [11] Chethan Pandarinath, Daniel J O’Shea, Jasmine Collins, Rafal Jozefowicz, Sergey D Stavisky,
460 Jonathan C Kao, Eric M Trautmann, Matthew T Kaufman, Stephen I Ryu, Leigh R Hochberg,
461 et al. Inferring single-trial neural population dynamics using sequential auto-encoders. *Nature
methods*, page 1, 2018.
- 463 [12] Alexander B Wiltschko, Matthew J Johnson, Giuliano Iurilli, Ralph E Peterson, Jesse M
464 Katon, Stan L Pashkovski, Victoria E Abraira, Ryan P Adams, and Sandeep Robert Datta.
465 Mapping sub-second structure in mouse behavior. *Neuron*, 88(6):1121–1135, 2015.
- 466 [13] Matthew J Johnson, David K Duvenaud, Alex Wiltschko, Ryan P Adams, and Sandeep R
467 Datta. Composing graphical models with neural networks for structured representations and
468 fast inference. In *Advances in neural information processing systems*, pages 2946–2954, 2016.
- 469 [14] Eleanor Batty, Matthew Whiteway, Shreya Saxena, Dan Biderman, Taiga Abe, Simon Musall,
470 Winthrop Gillis, Jeffrey Markowitz, Anne Churchland, John Cunningham, et al. Behavenet:
471 nonlinear embedding and bayesian neural decoding of behavioral videos. *Advances in Neural
472 Information Processing Systems*, 2019.
- 473 [15] Liam Paninski and John P Cunningham. Neural data science: accelerating the experiment-
474 analysis-theory cycle in large-scale neuroscience. *Current opinion in neurobiology*, 50:232–241,
475 2018.
- 476 [16] Mark K Transtrum, Benjamin B Machta, Kevin S Brown, Bryan C Daniels, Christopher R
477 Myers, and James P Sethna. Perspective: Sloppiness and emergent theories in physics, biology,
478 and beyond. *The Journal of chemical physics*, 143(1):07B201_1, 2015.
- 479 [17] Danilo Jimenez Rezende and Shakir Mohamed. Variational inference with normalizing flows.
480 *International Conference on Machine Learning*, 2015.
- 481 [18] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp.
482 *arXiv preprint arXiv:1605.08803*, 2016.
- 483 [19] George Papamakarios, Theo Pavlakou, and Iain Murray. Masked autoregressive flow for density
484 estimation. In *Advances in Neural Information Processing Systems*, pages 2338–2347, 2017.
- 485 [20] Gabriel Loaiza-Ganem, Yuanjun Gao, and John P Cunningham. Maximum entropy flow
486 networks. *International Conference on Learning Representations*, 2017.

- [21] Dustin Tran, Rajesh Ranganath, and David Blei. Hierarchical implicit models and likelihood-free variational inference. In *Advances in Neural Information Processing Systems*, pages 5523–5533, 2017.
- [22] Mark S Goldman, Jorge Golowasch, Eve Marder, and LF Abbott. Global structure, robustness, and modulation of neuronal models. *Journal of Neuroscience*, 21(14):5229–5238, 2001.
- [23] Gabrielle J Gutierrez, Timothy O’Leary, and Eve Marder. Multiple mechanisms switch an electrically coupled, synaptically inhibited neuron between competing rhythmic oscillators. *Neuron*, 77(5):845–858, 2013.
- [24] Ashok Litwin-Kumar, Robert Rosenbaum, and Brent Doiron. Inhibitory stabilization and visual coding in cortical circuits with multiple interneuron subtypes. *Journal of neurophysiology*, 115(3):1399–1409, 2016.
- [25] Chunyu A Duan, Marino Pagan, Alex T Piet, Charles D Kopec, Athena Akrami, Alexander J Riordan, Jeffrey C Erlich, and Carlos D Brody. Collicular circuits for flexible sensorimotor routing. *bioRxiv*, page 245613, 2018.
- [26] Francesca Mastrogiovanni and Srdjan Ostojic. Linking connectivity, dynamics, and computations in low-rank recurrent neural networks. *Neuron*, 99(3):609–623, 2018.
- [27] Sean R Bittner, Agostina Palmigiano, Kenneth D Miller, and John P Cunningham. Degenerate solution networks for theoretical neuroscience. *Computational and Systems Neuroscience Meeting (COSYNE), Lisbon, Portugal*, 2019.
- [28] Sean R Bittner, Alex T Piet, Chunyu A Duan, Agostina Palmigiano, Kenneth D Miller, Carlos D Brody, and John P Cunningham. Examining models in theoretical neuroscience with degenerate solution networks. *Bernstein Conference 2019, Berlin, Germany*, 2019.
- [29] Marcel Nonnenmacher, Pedro J Goncalves, Giacomo Bassetto, Jan-Matthis Lueckmann, and Jakob H Macke. Robust statistical inference for simulation-based models in neuroscience. In *Bernstein Conference 2018, Berlin, Germany*, 2018.
- [30] Deistler Michael, , Pedro J Goncalves, Kaan Oecal, and Jakob H Macke. Statistical inference for analyzing sloppiness in neuroscience models. In *Bernstein Conference 2019, Berlin, Germany*, 2019.

- 515 [31] Jan-Matthis Lueckmann, Pedro J Goncalves, Giacomo Bassetto, Kaan Öcal, Marcel Nonnenmacher, and Jakob H Macke. Flexible statistical inference for mechanistic models of neural dynamics. In *Advances in Neural Information Processing Systems*, pages 1289–1299, 2017.
- 518 [32] Eve Marder and Vatsala Thirumalai. Cellular, synaptic and network effects of neuromodulation. *Neural Networks*, 15(4-6):479–493, 2002.
- 520 [33] Astrid A Prinz, Dirk Bucher, and Eve Marder. Similar network activity from disparate circuit parameters. *Nature neuroscience*, 7(12):1345, 2004.
- 522 [34] Edwin T Jaynes. Information theory and statistical mechanics. *Physical review*, 106(4):620, 1957.
- 524 [35] Gamaleldin F Elsayed and John P Cunningham. Structure in neural population recordings: an expected byproduct of simpler phenomena? *Nature neuroscience*, 20(9):1310, 2017.
- 526 [36] Cristina Savin and Gašper Tkačik. Maximum entropy models as a tool for building precise neural controls. *Current opinion in neurobiology*, 46:120–126, 2017.
- 528 [37] Brendan K Murphy and Kenneth D Miller. Balanced amplification: a new mechanism of selective amplification of neural activity patterns. *Neuron*, 61(4):635–648, 2009.
- 530 [38] Hirofumi Ozeki, Ian M Finn, Evan S Schaffer, Kenneth D Miller, and David Ferster. Inhibitory stabilization of the cortical network underlies visual surround suppression. *Neuron*, 62(4):578–592, 2009.
- 533 [39] Daniel B Rubin, Stephen D Van Hooser, and Kenneth D Miller. The stabilized supralinear network: a unifying circuit motif underlying multi-input integration in sensory cortex. *Neuron*, 85(2):402–417, 2015.
- 536 [40] Henry Markram, Maria Toledo-Rodriguez, Yun Wang, Anirudh Gupta, Gilad Silberberg, and Caizhi Wu. Interneurons of the neocortical inhibitory system. *Nature reviews neuroscience*, 5(10):793, 2004.
- 539 [41] Bernardo Rudy, Gordon Fishell, SooHyun Lee, and Jens Hjerling-Leffler. Three groups of interneurons account for nearly 100% of neocortical gabaergic neurons. *Developmental neurobiology*, 71(1):45–61, 2011.
- 542 [42] Robin Tremblay, Soohyun Lee, and Bernardo Rudy. GABAergic Interneurons in the Neocortex: From Cellular Properties to Circuits. *Neuron*, 91(2):260–292, 2016.

- 544 [43] Carsten K Pfeffer, Mingshan Xue, Miao He, Z Josh Huang, and Massimo Scanziani. Inhibition
545 of inhibition in visual cortex: the logic of connections between molecularly distinct
546 interneurons. *Nature Neuroscience*, 16(8):1068, 2013.
- 547 [44] Luis Carlos Garcia Del Molino, Guangyu Robert Yang, Jorge F. Mejias, and Xiao Jing Wang.
548 Paradoxical response reversal of top- down modulation in cortical circuits with three interneu-
549 ron types. *Elife*, 6:1–15, 2017.
- 550 [45] Guang Chen, Carl Van Vreeswijk, David Hansel, and David Hansel. Mechanisms underlying
551 the response of mouse cortical networks to optogenetic manipulation. 2019.
- 552 [46] (2018) Allen Institute for Brain Science. Layer 4 model of v1. available from:
553 <https://portal.brain-map.org/explore/models/l4-mv1>.
- 554 [47] Yazan N Billeh, Binghuang Cai, Sergey L Gratiy, Kael Dai, Ramakrishnan Iyer, Nathan W
555 Gouwens, Reza Abbasi-Asl, Xiaoxuan Jia, Joshua H Siegle, Shawn R Olsen, et al. Systematic
556 integration of structural and functional data into multi-scale models of mouse primary visual
557 cortex. *bioRxiv*, page 662189, 2019.
- 558 [48] Chunyu A Duan, Jeffrey C Erlich, and Carlos D Brody. Requirement of prefrontal and midbrain
559 regions for rapid executive control of behavior in the rat. *Neuron*, 86(6):1491–1503, 2015.
- 560 [49] Omri Barak. Recurrent neural networks as versatile tools of neuroscience research. *Current*
561 *opinion in neurobiology*, 46:1–6, 2017.
- 562 [50] David Sussillo and Omri Barak. Opening the black box: low-dimensional dynamics in high-
563 dimensional recurrent neural networks. *Neural computation*, 25(3):626–649, 2013.
- 564 [51] Rodrigo Echeveste, Laurence Aitchison, Guillaume Hennequin, and Máté Lengyel. Cortical-like
565 dynamics in recurrent circuits optimized for sampling-based probabilistic inference. *bioRxiv*,
566 page 696088, 2019.
- 567 [52] Robert E Kass and Valérie Ventura. A spike-train probability model. *Neural computation*,
568 13(8):1713–1720, 2001.
- 569 [53] Emery N Brown, Loren M Frank, Dengda Tang, Michael C Quirk, and Matthew A Wilson.
570 A statistical paradigm for neural spike train decoding applied to position prediction from
571 ensemble firing patterns of rat hippocampal place cells. *Journal of Neuroscience*, 18(18):7411–
572 7425, 1998.

- 573 [54] Liam Paninski. Maximum likelihood estimation of cascade point-process neural encoding
 574 models. *Network: Computation in Neural Systems*, 15(4):243–262, 2004.
- 575 [55] M Yu Byron, John P Cunningham, Gopal Santhanam, Stephen I Ryu, Krishna V Shenoy, and
 576 Maneesh Sahani. Gaussian-process factor analysis for low-dimensional single-trial analysis
 577 of neural population activity. In *Advances in neural information processing systems*, pages
 578 1881–1888, 2009.
- 579 [56] Kenneth W Latimer, Jacob L Yates, Miriam LR Meister, Alexander C Huk, and Jonathan W
 580 Pillow. Single-trial spike trains in parietal cortex reveal discrete steps during decision-making.
 581 *Science*, 349(6244):184–187, 2015.
- 582 [57] Lea Duncker, Gergo Bohner, Julien Boussard, and Maneesh Sahani. Learning interpretable
 583 continuous-time models of latent stochastic dynamical systems. *Proceedings of the 36th Inter-*
 584 *national Conference on Machine Learning*, 2019.
- 585 [58] Steven H Strogatz. Nonlinear dynamics and chaos: with applications to physics. *Biology,*
 586 *Chemistry, and Engineering (Studies in Nonlinearity)*, Perseus, Cambridge, UK, 1994.
- 587 [59] Rajesh Ranganath, Sean Gerrish, and David Blei. Black box variational inference. In *Artificial*
 588 *Intelligence and Statistics*, pages 814–822, 2014.
- 589 [60] Martin J Wainwright, Michael I Jordan, et al. Graphical models, exponential families, and
 590 variational inference. *Foundations and Trends® in Machine Learning*, 1(1–2):1–305, 2008.
- 591 [61] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International*
 592 *Conference on Learning Representations*, 2015.
- 593 [62] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp.
 594 *Proceedings of the 5th International Conference on Learning Representations*, 2017.
- 595 [63] David M Blei, Alp Kucukelbir, and Jon D McAuliffe. Variational inference: A review for
 596 statisticians. *Journal of the American Statistical Association*, 112(518):859–877, 2017.

597 **A Acknowledgements**

598 This work was funded by NSF Graduate Research Fellowship, DGE-1644869, McKnight Endow-
 599 ment Fund, NIH NINDS 5R01NS100066, Simons Foundation 542963, NSF NeuroNex Award DBI-
 600 1707398, and The Gatsby Charitable Foundation. Helpful conversations were had with Francesca

601 Mastrogiuseppe, Srdjan Ostojic, James Fitzgerald, Stephen Baccus, Dhruva Raman, Mehrdad
 602 Jazayeri, Liam Paninski, and Larry Abbott.

603 **B Methods**

604 **B.1 Emergent property inference (EPI)**

605 Emergent property inference (EPI) learns distributions of theoretical model parameters that pro-
 606 duce emergent properties of interest by combining ideas from maximum entropy flow networks
 607 (MEFNs) [20] and likelihood-free variational inference (LFVI) [21]. Consider model parameteri-
 608 zation z and data x which has an intractable likelihood $p(x | z)$ defined by a model simulator of
 609 which samples are available $x \sim p(x | z)$. EPI optimizes a distribution $q_\theta(z)$ (itself parameterized
 610 by θ) of model parameters z to produce an emergent property of interest \mathcal{B} ,

$$\mathcal{B} \triangleq \mathbb{E}_{z \sim q_\theta} [\mathbb{E}_{x \sim p(x|z)} [T(x)]] = \mu \quad (15)$$

611 Precisely, over the EPI distribution of parameters $q_\theta(z)$ and distribution of simulated activity
 612 $p(x | z)$, the emergent property statistics $T(x)$ must equal the emergent property values μ on
 613 average. This is a viable way to represent emergent properties in theoretical models, as we have
 614 demonstrated in the main text, and enables the EPI optimization.

615 With EPI, we use deep probability distributions to learn flexible approximations to model parameter
 616 distributions $q_\theta(z)$. In deep probability distributions, a simple random variable $w \sim q_0(w)$ is
 617 mapped deterministically via a sequence of deep neural network layers (f_1, \dots, f_l) parameterized by
 618 weights and biases θ to the support of the distribution of interest:

$$z = f_\theta(\omega) = f_l(\dots f_1(w)) \quad (16)$$

619 Given a simulator defined by a theoretical model $x \sim p(x | z)$ and some emergent property of
 620 interest \mathcal{B} , $q_\theta(z)$ is optimized via the neural network parameters θ to find an optimally entropic
 621 distribution q_θ^* within the deep variational family \mathcal{Q} producing the emergent property:

$$\begin{aligned} q_\theta^*(z) &= \operatorname{argmax}_{q_\theta \in \mathcal{Q}} H(q_\theta(z)) \\ \text{s.t. } \mathbb{E}_{z \sim q_\theta} [\mathbb{E}_{x \sim p(x|z)} [T(x)]] &= \mu \end{aligned} \quad (17)$$

622 Since we are optimizing parameters θ of our deep probability distribution with respect to the entropy
 623 $H(q_\theta(z))$, we will need to take gradients with respect to the log probability density of samples from

624 the deep probability distribution.

$$H(q_\theta(z)) = \int -q_\theta(z) \log(q_\theta(z)) dz = \mathbb{E}_{z \sim q_\theta} [-\log(q_\theta(z))] = \mathbb{E}_{w \sim q_0} [-\log(q_\theta(f_\theta(w)))] \quad (18)$$

625

$$\nabla_\theta H(q_\theta(z)) = \mathbb{E}_{w \sim q_0} [-\nabla_\theta \log(q_\theta(f_\theta(w)))] \quad (19)$$

626 This optimization is done using the approach of MEFN [20], using architectures for deep probability
 627 distributions, called normalizing flows (see Section B.1.3), conferring a tractable calculation of
 628 sample log probability. In EPI, this methodology for learning maximum entropy distributions is
 629 repurposed toward variational learning of model parameter distributions. Similar to LFVI [21], we
 630 are motivated to do variational learning in models with intractable likelihood functions, in which
 631 standard methods like stochastic gradient variational Bayes [6] or black box variational inference[59]
 632 are not tractable. Furthermore, EPI focuses on setting mathematically defined emergent property
 633 statistics to emergent property values of interest, whereas LFVI is focused on learning directly from
 634 datasets. Optimizing this objective is a technological challenge, the details of which we elaborate
 635 in Section B.1.2. Before going through those details, we ground this optimization in a toy example.

636 **B.1.1 Example: 2D LDS**

637 To gain intuition for EPI, consider a two-dimensional linear dynamical system model:

$$\tau \frac{dx}{dt} = Ax \quad (20)$$

638 with

$$A = \begin{bmatrix} a_1 & a_2 \\ a_3 & a_4 \end{bmatrix} \quad (21)$$

639 To run EPI with the dynamics matrix elements as the free parameters $z = [a_1 \ a_2 \ a_3 \ a_4]$ (fixing
 640 $\tau = 1$), the emergent property statistics $T(x)$ were chosen to contain the first- and second-moments
 641 of the oscillatory frequency $2\pi\text{imag}(\lambda_1)$ and the growth/decay factor $\text{real}(\lambda_1)$ of the oscillating
 642 system. λ_1 is the eigenvalue of greatest real part when the imaginary component is zero, and
 643 alternatively of positive imaginary component when the eigenvalues are complex conjugate pairs.
 644 To learn the distribution of real entries of A that produce a band of oscillating systems around
 645 1Hz, we formalized this emergent property as $\text{real}(\lambda_1)$ having mean zero with variance 0.25², and

646 the oscillation frequency $2\pi\text{imag}(\lambda_1)$ having mean $\omega = 1$ Hz with variance $(0.1\text{Hz})^2$:

$$\mathbb{E}[T(x)] \triangleq \mathbb{E} \begin{bmatrix} \text{real}(\lambda_1) \\ \text{imag}(\lambda_1) \\ (\text{real}(\lambda_1) - 0)^2 \\ (\text{imag}(\lambda_1) - 2\pi\omega)^2 \end{bmatrix} = \begin{bmatrix} 0.0 \\ 2\pi\omega \\ 0.25^2 \\ (2\pi 0.1)^2 \end{bmatrix} \triangleq \mu \quad (22)$$

647

648 Unlike the models we presented in the main text, this model admits an analytical form for the the
 649 mean emergent property statistics given parameter z , since the eigenvalues can be calculated using
 650 the quadratic formula:

$$\lambda = \frac{\left(\frac{a_1+a_4}{\tau}\right) \pm \sqrt{\left(\frac{a_1+a_4}{\tau}\right)^2 + 4\left(\frac{a_2a_3-a_1a_4}{\tau}\right)}}{2} \quad (23)$$

651 Importantly, even though $\mathbb{E}_{x \sim p(x|z)}[T(x)]$ is calculable directly via a closed form function and does
 652 not require simulation, we cannot derive the distribution q_θ^* directly. This is due to the formally hard
 653 problem of the backward mapping: finding the natural parameters η from the mean parameters μ
 654 of an exponential family distribution [60]. Instead, we used EPI to approximate this distribution
 655 (Fig. S1B). We used a real-NVP normalizing flow architecture with four masks, two neural network
 656 layers of 15 units per mask, with batch normalization momentum 0.99, mapped onto a support of
 657 $z_i \in [-20, 20]$. (see Section B.1.3).

658 Even this relatively simple system has nontrivial (though intuitively sensible) structure in the
 659 parameter distribution. To validate our method (further than that of the underlying technology on
 660 a ground truth solution [20]) we analytically derived the contours of the probability density from the
 661 emergent property statistics and values (Fig. S2). In the $a_1 - a_4$ plane, the black line at $\text{real}(\lambda_1) =$
 662 $\frac{a_1+a_4}{2} = 0$, and the dotted black line at the standard deviation $\text{real}(\lambda_1) = \frac{a_1+a_4}{2} \pm 0.25$, and the gray
 663 line at twice the standard deviation $\text{real}(\lambda_1) = \frac{a_1+a_4}{2} \pm 0.5$ follow the contour of probability density
 664 of the samples. (Fig. 2A). The distribution precisely reflects the desired statistical constraints and
 665 model degeneracy in the sum of a_1 and a_4 . Intuitively, the parameters equivalent with respect to
 666 emergent property statistic $\text{real}(\lambda_1)$ have similar log densities.

667 To explain the bimodality of the EPI distribution, we examined the imaginary component of λ_1 .

668 When $\text{real}(\lambda_1) = \frac{a_1+a_4}{2} = 0$, we have

$$\text{imag}(\lambda_1) = \begin{cases} \sqrt{\frac{a_1a_4-a_2a_3}{\tau}}, & \text{if } a_1a_4 < a_2a_3 \\ 0 & \text{otherwise} \end{cases} \quad (24)$$

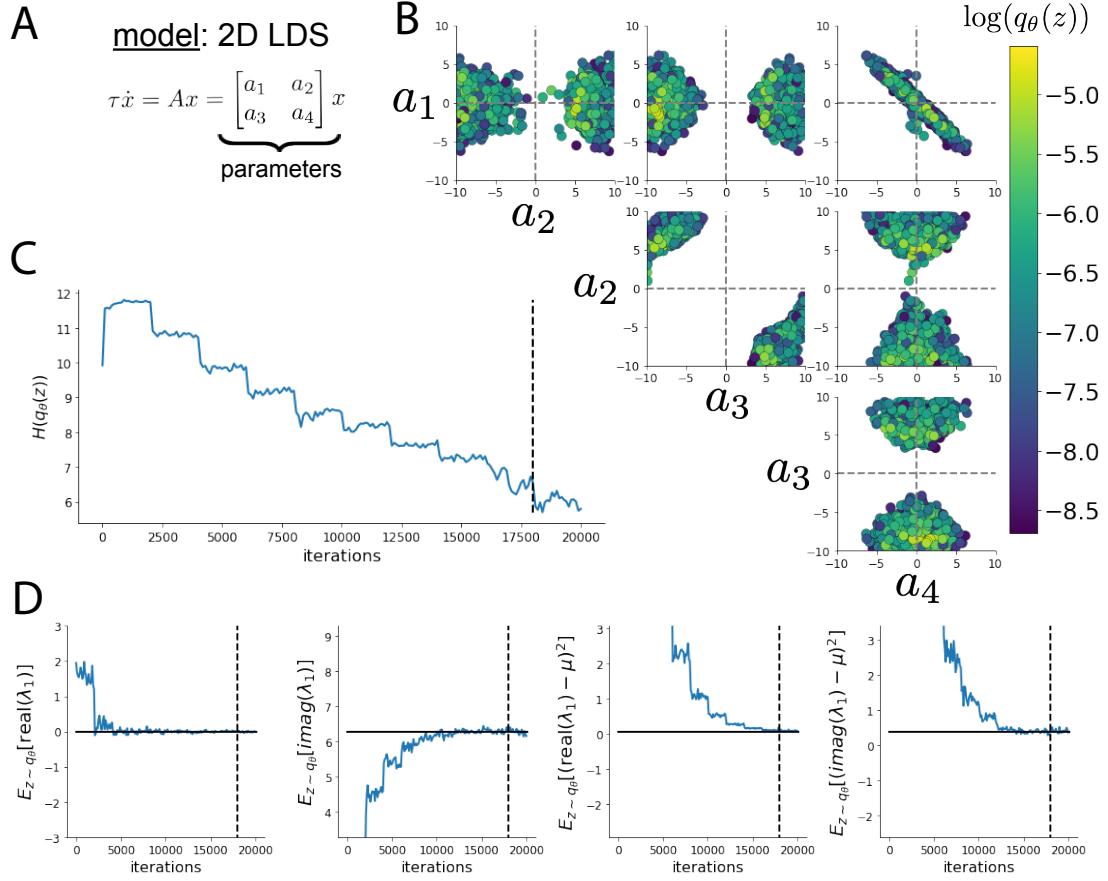


Fig. S1: A. Two-dimensional linear dynamical system model, where real entries of the dynamics matrix A are the parameters. B. The DSN distribution for a two-dimensional linear dynamical system with $\tau = 1$ that produces an average of 1Hz oscillations with some small amount of variance. C. Entropy throughout the optimization. At the beginning of each augmented Lagrangian epoch (2,000 iterations), the entropy dipped due to the shifted optimization manifold where emergent property constraint satisfaction is increasingly weighted. D. Emergent property moments throughout optimization. At the beginning of each augmented Lagrangian epoch, the emergent property moments adjust closer to their constraints.

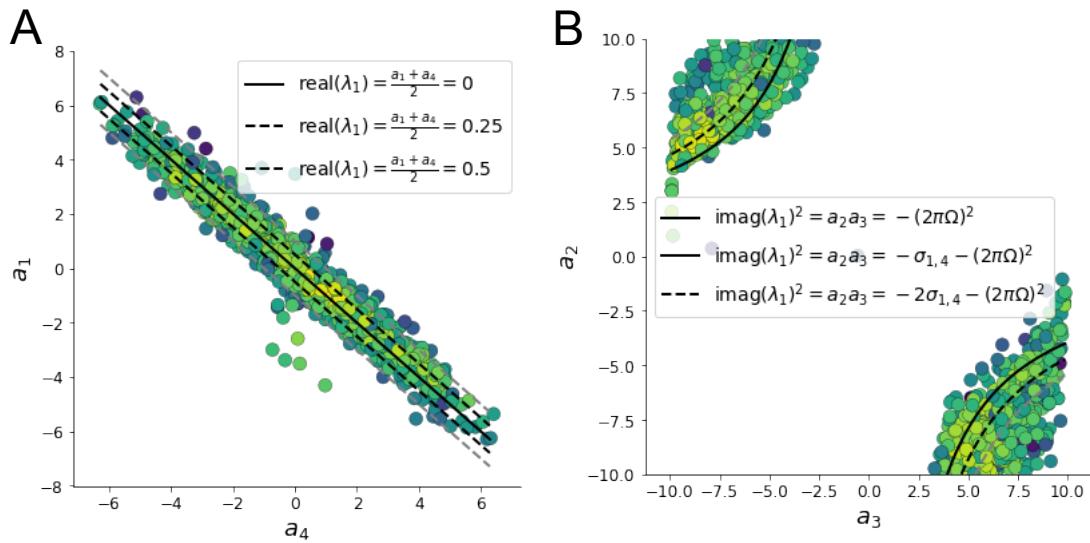


Fig. S2: A. Probability contours in the $a_1 - a_4$ plane can be derived from the relationship to emergent property statistic of growth/decay factor $\text{real}(\lambda_1)$. B. Probability contours in the $a_2 - a_3$ plane can be derived from the emergent property statistic of oscillation frequency $2\pi\text{imag}(\lambda_1)$ (see text).

669 When $\tau = 1$ and $a_1 a_4 > a_2 a_3$ (center of distribution above), we have the following equation for the
670 other two dimensions:

$$\text{imag}(\lambda_1)^2 = a_1 a_4 - a_2 a_3 \quad (25)$$

671 Since we constrained $\mathbb{E}_{z \sim q_\theta} [\text{imag}(\lambda)] = 2\pi$ (with $\omega = 1$), we can plot contours of the equation
672 $\text{imag}(\lambda_1)^2 = a_1 a_4 - a_2 a_3 = (2\pi)^2$ for various $a_1 a_4$ (Fig. S2A). If $\sigma_{1,4} = \mathbb{E}_{z \sim q_\theta} (|a_1 a_4 - E_{q_\theta}[a_1 a_4]|)$,
673 then we plot the contours as $a_1 a_4 = 0$ (black), $a_1 a_4 = -\sigma_{1,4}$ (black dotted), and $a_1 a_4 = -2\sigma_{1,4}$
674 (grey dotted) (Fig. S2B). This validates the curved structure of the inferred distribution learned
675 through EPI. We take steps in negative standard deviation of $a_1 a_4$ (dotted and gray lines), since
676 there are few positive values $a_1 a_4$ in the learned distribution. Subtler combinations of model and
677 emergent property will have more complexity, further motivating the use of EPI for understanding
678 these systems. As we expect, the distribution results in samples of two-dimensional linear systems
679 oscillating near 1Hz (Fig. S3).

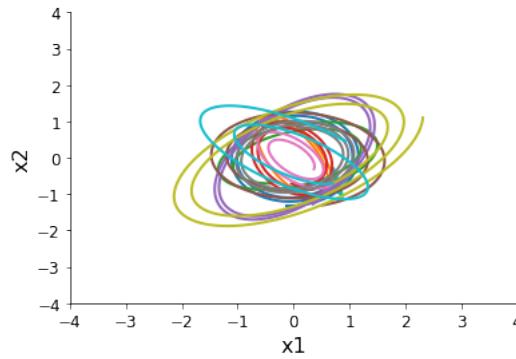


Fig. S3: Sampled dynamical system trajectories from the EPI distribution. Each trajectory is initialized at $x(0) = \begin{bmatrix} \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} \end{bmatrix}$.

680 B.1.2 Augmented Lagrangian optimization

681 To optimize $q_\theta(z)$ in Equation 17, the constrained optimization is executed using the augmented
 682 Lagrangian method. The following objective is minimized:

$$L(\theta; \eta, c) = -H(q_\theta) + \eta^\top R(\theta) + \frac{c}{2} \|R(\theta)\|^2 \quad (26)$$

683 where $R(\theta) = \mathbb{E}_{z \sim q_\theta} [\mathbb{E}_{x \sim p(x|z)} [T(x) - \mu]]$, $\eta \in \mathbb{R}^m$ are the Lagrange multipliers where $m = |\mu| = |T(x)|$, and c is the penalty coefficient. These Lagrange multipliers are closely related to the natural
 684 parameters of exponential families (see Section B.1.4). Deep neural network weights and biases θ of
 685 the deep probability distribution are optimized according to Equation 26 using the Adam optimizer
 686 with its standard parameterization [61]. η is initialized to the zero vector and adapted following
 687 each augmented Lagrangian epoch, which is a period of optimization with fixed (η, c) for a given
 688 number of stochastic optimization iterations. A low value of c is used initially, and conditionally
 689 increased after each epoch based on constraint error reduction. For example, the initial value of
 690 c was $c_0 = 10^{-3}$ during EPI with the linear two-dimensional system (Fig. S1C). The penalty
 691 coefficient is updated based on the result of a hypothesis test regarding the reduction in constraint
 692 violation. The p-value of $\mathbb{E}[\|R(\theta_{k+1})\|] > \gamma \mathbb{E}[\|R(\theta_k)\|]$ is computed, and c_{k+1} is updated to βc_k
 693 with probability $1 - p$. The other update rule is $\eta_{k+1} = \eta_k + c_k \frac{1}{n} \sum_{i=1}^n (T(x^{(i)}) - \mu)$ given a batch
 694 size n . Throughout the study, $\beta = 4.0$, $\gamma = 0.25$, and the batch size was a hyperparameter, which
 695 varied according to the application of EPI.

696 The intention is that c and η start at values encouraging entropic growth early in optimization.
 697 Then, as they increase in magnitude with each training epoch, the constraint satisfaction terms

699 are increasingly weighted, resulting in a decrease in entropy. This encourages the discovery of
 700 suitable regions of parameter space, and the subsequent refinement of the distribution to produce
 701 the emergent property. In the two-dimensional example, each augmented Lagrangian epoch ran for
 702 2,000 iterations (Fig. S1C-D). Notice the initial entropic growth, and subsequent reduction upon
 703 each update of η and c . The momentum parameters of the Adam optimizer were reset at the end
 704 of each augmented Lagrangian epoch.

705 Rather than starting optimization from some drawn θ from a glorot uniform or other distribu-
 706 tion, we found that initializing $q_\theta(z)$ to approximate an isotropic Gaussian distribution conferred
 707 more stable, consistent optimization. The parameters of the initialization Gaussain were chosen
 708 on an application-specific basis. Throughout the study, we chose isotropic Gaussians with mean
 709 μ_{init} at the center of the distribution support and some standard deviation σ_{init} , except when we
 710 demonstrate how to use grid search to inform the initialization in Section B.2.2.

711 To assess whether EPI distribution $q_\theta(z)$ produces the emergent property, we defined a hypothesis
 712 testing convergence criteria. The algorithm has converged when a null hypothesis test of constraint
 713 violations $R(\theta)_i$ being zero is accepted for all constraints $i \in \{1, \dots, m\}$ at a significance threshold
 714 $\alpha = 0.05$. This significance threshold is adjusted through Bonferroni correction according to the
 715 number of constraints m . The p-values for each constraint are calculated according to a two-
 716 tailed nonparameteric test, where 200 estimations of the sample mean $R(\theta)^i$ are made from k
 717 resamplings of z of a finite sample of size n taken at the end of the augmented Lagrangian epoch.
 718 k is determined by a fraction of the batch size ν , which varies according to the application. In the
 719 linear two-dimensional system example, we used a batch size of $n = 1000$ and set $\nu = 0.1$ resulting
 720 in convergence after the ninth epoch of optimization. (Fig. S1C-D black dotted line).

721 B.1.3 Normalizing flows

722 Deep probability models typically consist of several layers of fully connected neural networks.
 723 When each neural network layer is restricted to be a bijective function, the sample density can be
 724 calculated using the change of variables formula at each layer of the network. For $z' = f(z)$,

$$q(z') = q(f^{-1}(z')) \left| \det \frac{\partial f^{-1}(z')}{\partial z'} \right| = q(z) \left| \det \frac{\partial f(z)}{\partial z} \right|^{-1} \quad (27)$$

725 However, this computation has cubic complexity in dimensionality for fully connected layers. By
 726 restricting our layers to normalizing flows [17] – bijective functions with fast log determinant Ja-

727 cobian computations, we can tractably optimize deep generative models with objectives that are a
 728 function of sample density, like entropy. Most of our analyses use either a planar flow [17] or real
 729 NVP [62], which have proven effective in our architecture searches. Planar flow architectures are
 730 specified by the number of planar bijection layers used, while real NVP architectures are specified
 731 by the number of masks, neural network layers per mask, units per layer, and batch normalization
 732 momentum parameter.

733 **B.1.4 Emergent property inference as variational inference in an exponential family**

734 Consider the goal of doing variational inference with an exponential family posterior distribution
 735 $p(z | x)$. We use the following abbreviated notation to collect the base measure $b(z)$ and sufficient
 736 statistics $T(z)$ into $\tilde{T}(z)$ and likewise concatenate a 1 onto the end of the natural parameter $\tilde{\eta}(x)$.
 737 The log normalizing constant $A(\eta(x))$ remains unchanged.

$$\begin{aligned} p(z | x) &= b(z) \exp \left(\eta(x)^\top T(z) - A(\eta(x)) \right) = \exp \left(\begin{bmatrix} \eta(x) \\ 1 \end{bmatrix}^\top \begin{bmatrix} T(z) \\ b(z) \end{bmatrix} - A(\eta(x)) \right) \\ &= \exp \left(\tilde{\eta}(x)^\top \tilde{T}(z) - A(\eta(x)) \right) \end{aligned} \quad (28)$$

738 Variational inference with an exponential family posterior distribution uses optimization to mini-
 739 mize the following divergence [63]:

$$q_\theta^* = \underset{q_\theta \in Q}{\operatorname{argmin}} KL(q_\theta || p(z | x)) \quad (29)$$

740 $q_\theta(z)$ is the variational approximation to the posterior with variational parameters θ . We can write
 741 this KL divergence in terms of entropy of the variational approximation.

$$KL(q_\theta || p(z | x)) = \mathbb{E}_{z \sim q_\theta} [\log(q_\theta(z))] - \mathbb{E}_{z \sim q_\theta} [\log(p(z | x))] \quad (30)$$

$$= -H(q_\theta) - \mathbb{E}_{z \sim q_\theta} [\tilde{\eta}(x)^\top \tilde{T}(z) - A(\eta(x))] \quad (31)$$

742 743 As far as the variational optimization is concerned, the log normalizing constant is independent of
 744 $q_\theta(z)$, so it can be dropped.

$$\underset{q_\theta \in Q}{\operatorname{argmin}} KL(q_\theta || p(z | x)) = \underset{q_\theta \in Q}{\operatorname{argmin}} -H(q_\theta) - \mathbb{E}_{z \sim q_\theta} [\tilde{\eta}(x)^\top \tilde{T}(z)] \quad (32)$$

745 Further, we can write the objective in terms of the first moment of the sufficient statistics $\mu =$
 746 $\mathbb{E}_{z \sim p(z|x)} [T(z)]$.

$$= \underset{q_\theta \in Q}{\operatorname{argmin}} -H(q_\theta) - \mathbb{E}_{z \sim q_\theta} [\tilde{\eta}(x)^\top (\tilde{T}(z) - \mu)] + \tilde{\eta}(x)^\top \mu \quad (33)$$

747

$$= \operatorname{argmin}_{q_\theta \in Q} -H(q_\theta) - \mathbb{E}_{z \sim q_\theta} \left[\tilde{\eta}(x)^\top (\tilde{T}(z) - \mu) \right] \quad (34)$$

748 In comparison, in emergent property inference (EPI), we're solving the following problem.

$$q_\theta^*(z) = \operatorname{argmax}_{q_\theta \in Q} H(q_\theta(z)), \text{ s.t. } \mathbb{E}_{z \sim q_\theta} [\mathbb{E}_{x \sim p(x|z)} [T(x)]] = \mu \quad (35)$$

749 The Lagrangian objective (without the augmentation) is

$$q_\theta^* = \operatorname{argmin}_{q_\theta \in Q} -H(q_\theta) + \eta_{\text{opt}}^\top \left(\mathbb{E}_{z \sim q_\theta} [\tilde{T}(z)] - \mu \right) \quad (36)$$

750 As the optimization proceeds, η_{opt}^\top should converge to the natural parameter $\tilde{\eta}(x)$ through its
751 adaptations in each epoch (see Section B.1.2).

752 The derivation of the natural parameter $\tilde{\eta}(x)$ of an exponential family distribution from its mean
753 parameter μ is referred to as the backward mapping and is formally hard to identify [60]. Since
754 this backward mapping is deterministic, we can replace the notation of $p(z | x)$ with $p(z | \mathcal{B})$
755 conceptualizing an inferred distribution that obeys emergent property \mathcal{B} (see Section B.1).

756 B.2 Theoretical models

757 In this study, we used emergent property inference to examine several models relevant to theoretical
758 neuroscience. Here, we provide the details of each model and the related analyses.

759 B.2.1 Stomatogastric ganglion

760 We analyze how the parameters $z = [g_{el} \ g_{synA}]$ govern the emergent phenomena of network
761 syncing in a model of the stomatogastric ganglion (STG) shown in Figure 1A with activity $x =$
762 $[x_{f1}, x_{f2}, x_{\text{hub}}, x_{s1}, x_{s2}]$. Each neuron's membrane potential $x_\alpha(t)$ for $\alpha \in \{f1, f2, \text{hub}, s1, s2\}$ is the
763 solution of the following differential equation:

$$C_m \frac{dx_\alpha}{dt} = -[h_{leak}(x; z) + h_{Ca}(x; z) + h_K(x; z) + h_{hyp}(x; z) + h_{elec}(x; z) + h_{syn}(x; z)] \quad (37)$$

764 The membrane potential of each neuron is affected by the leak, calcium, potassium, hyperpolariza-
765 tion, electrical and synaptic currents, respectively, which are functions of all membrane potentials
766 and the conductance parameters z . The capacitance of the cell membrane was set to $C_m = 1nF$.
767 Specifically, the currents are the difference in the neuron's membrane potential and that current
768 type's reversal potential multiplied by a conductance:

$$h_{leak}(x; z) = g_{leak}(x_\alpha - V_{leak}) \quad (38)$$

769

$$h_{elec}(x; z) = g_{el}(x_\alpha^{post} - x_\alpha^{pre}) \quad (39)$$

770

$$h_{syn}(x; z) = g_{syn}S_\infty^{pre}(x_\alpha^{post} - V_{syn}) \quad (40)$$

771

$$h_{Ca}(x; z) = g_{Ca}M_\infty(x_\alpha - V_{Ca}) \quad (41)$$

772

$$h_K(x; z) = g_KN(x_\alpha - V_K) \quad (42)$$

773

$$h_{hyp}(x; z) = g_hH(x_\alpha - V_{hyp}) \quad (43)$$

774 The reversal potentials were set to $V_{leak} = -40mV$, $V_{Ca} = 100mV$, $V_K = -80mV$, $V_{hyp} = -20mV$,
 775 and $V_{syn} = -75mV$. The other conductance parameters were fixed to $g_{leak} = 1 \times 10^{-4}\mu S$. g_{Ca} ,
 776 g_K , and g_{hyp} had different values based on fast, intermediate (hub) or slow neuron. Fast: $g_{Ca} =$
 777 1.9×10^{-2} , $g_K = 3.9 \times 10^{-2}$, and $g_{hyp} = 2.5 \times 10^{-2}$. Intermediate: $g_{Ca} = 1.7 \times 10^{-2}$, $g_K = 1.9 \times 10^{-2}$,
 778 and $g_{hyp} = 8.0 \times 10^{-3}$. Intermediate: $g_{Ca} = 8.5 \times 10^{-3}$, $g_K = 1.5 \times 10^{-2}$, and $g_{hyp} = 1.0 \times 10^{-2}$.

779 Furthermore, the Calcium, Potassium, and hyperpolarization channels have time-dependent gating
 780 dynamics dependent on steady-state gating variables M_∞ , N_∞ and H_∞ , respectively.

$$M_\infty = 0.5 \left(1 + \tanh \left(\frac{x_\alpha - v_1}{v_2} \right) \right) \quad (44)$$

781

$$\frac{dN}{dt} = \lambda_N(N_\infty - N) \quad (45)$$

782

$$N_\infty = 0.5 \left(1 + \tanh \left(\frac{x_\alpha - v_3}{v_4} \right) \right) \quad (46)$$

783

$$\lambda_N = \phi_N \cosh \left(\frac{x_\alpha - v_3}{2v_4} \right) \quad (47)$$

784

$$\frac{dH}{dt} = \frac{(H_\infty - H)}{\tau_h} \quad (48)$$

785

$$H_\infty = \frac{1}{1 + \exp \left(\frac{x_\alpha + v_5}{v_6} \right)} \quad (49)$$

786

$$\tau_h = 272 - \left(\frac{-1499}{1 + \exp \left(\frac{-x_\alpha + v_7}{v_8} \right)} \right) \quad (50)$$

787 where we set $v_1 = 0mV$, $v_2 = 20mV$, $v_3 = 0mV$, $v_4 = 15mV$, $v_5 = 78.3mV$, $v_6 = 10.5mV$,
 788 $v_7 = -42.2mV$, $v_8 = 87.3mV$, $v_9 = 5mV$, and $v_{th} = -25mV$. These are the same parameter
 789 values used in [23].

790 Finally, there is a synaptic gating variable as well:

$$S_\infty = \frac{1}{1 + \exp \left(\frac{v_{th} - x_\alpha}{v_9} \right)} \quad (51)$$

791 When the dynamic gating variables are considered, this is actually a 15-dimensional nonlinear
 792 dynamical system.

793 In order to measure the frequency of the hub neuron during EPI, the STG model was simulated
 794 for $T = 200$ time steps of $dt = 25ms$. In EPI, since gradients are taken through the simulation
 795 process, the number of time steps are kept modest if possible. The chosen dt and T were the
 796 most computationally convenient choices yielding accurate frequency measurement. Poor resolution
 797 afforded by the discrete Fourier transform motivated the use of an alternative basis of complex
 798 exponentials to measure spiking frequency. Instead, we used a basis of complex exponentials with
 799 frequencies from 0.0-1.0 Hz at 0.01Hz resolution, $\Phi = [0.0, 0.01, \dots, 1.0]^\top$

800 Another consideration was that the frequency spectra of the neuron membrane potentials had sev-
 801 eral peaks. High-frequency sub-threshold activity obscured the maximum frequency measurement
 802 in the complex exponential basis. Accordingly, subthreshold activity was set to zero, and the
 803 whole signal was low-pass filtered with a moving average window of length 20. The signal was
 804 subsequently mean centered. After this pre-processing, the maximum frequency in the filter bank
 805 accurately reflected the firing frequency.

806 Finally, to differentiate through the maximum frequency identification, we used a sum-of-powers
 807 normalization. Let $\mathcal{X}_\alpha \in \mathcal{C}^{|\Phi|}$ be the complex exponential filter bank dot products with the signal
 808 $x_\alpha \in \mathbb{R}^N$, where $\alpha \in \{f1, f2, \text{hub}, s1, s2\}$. The “frequency identification” vector is

$$v_\alpha = \frac{|\mathcal{X}_\alpha|^\beta}{\sum_{k=1}^N |\mathcal{X}_\alpha(k)|^\beta} \quad (52)$$

809 The frequency is then calculated as $\omega_\alpha = v_\alpha^\top \Phi$ with $\beta = 100$.

810 Network syncing, like all other emergent properties in this work, are defined by the emergent
 811 property statistics and values. The emergent property statistics are the first- and second-moments
 812 of the firing frequencies. The first moments are set to 0.53Hz, while the second moments are set to

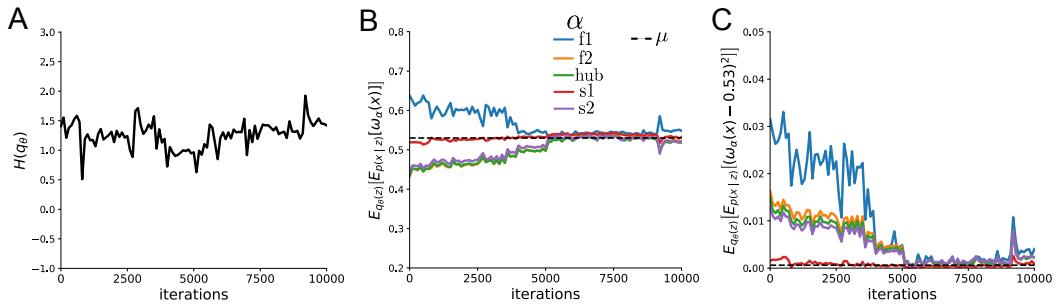


Fig. S4: Emergent property inference of the STG model producing network syncing. A. Entropy throughout optimization. B. The first moment emergent property statistics converge to the emergent property values at 10,000 iterations, following the fourth augmented Lagrangian epoch of 2,500 iterations. (There is no convergence at the end of the third epoch, because $q_\theta(z)$ failed to produce enough samples yielding $\omega_{f1}(x)$ less than 0.53Hz.) C. The second moment emergent property statistics converge to the emergent property values.

813 0.025Hz²:

$$E \begin{bmatrix} \omega_{f1} \\ \omega_{f2} \\ \omega_{hub} \\ \omega_{s1} \\ \omega_{s2} \\ (\omega_{f1} - 0.53)^2 \\ (\omega_{f2} - 0.53)^2 \\ (\omega_{hub} - 0.53)^2 \\ (\omega_{s1} - 0.53)^2 \\ (\omega_{s2} - 0.53)^2 \end{bmatrix} = \begin{bmatrix} 0.53 \\ 0.53 \\ 0.53 \\ 0.53 \\ 0.53 \\ 0.025^2 \\ 0.025^2 \\ 0.025^2 \\ 0.025^2 \\ 0.025^2 \end{bmatrix} \quad (53)$$

814 Throughout optimization, the augmented Lagrangian parameters η and c , were updated after each
 815 epoch of 2,500 iterations (see Section B.1.2). The optimization converged after four epochs (Fig.
 816 S4).

817 For EPI in Fig 2C, we used a real NVP architecture with four masks and two layers of 10 units per
 818 mask, and batch normalization momentum of 0.99 mapped onto a support of $z \in \left[\begin{bmatrix} 4 & 0 \end{bmatrix}, \begin{bmatrix} 8 & 4 \end{bmatrix} \right]$.
 819 We used an augmented Lagrangian coefficient of $c_0 = 10^2$, a batch size $n = 300$, set $\nu = 0.1$, and
 820 initialized $q_\theta(z)$ to produce an isotropic Gaussian with mean $\mu_{\text{init}} = \begin{bmatrix} 6 & 2 \end{bmatrix}$ with standard deviation
 821 $\sigma_{\text{init}} = 0.5$.

We calculated the Hessian at the mode of the inferred EPI distribution. The Hessian of a probability model is the second order gradient of the log probability density $\log q_\theta(z)$ with respect to the parameters z : $\frac{\partial^2 \log q_\theta(z)}{\partial z \partial z^\top}$. With EPI, we can examine the Hessian, which is analytically available throughout the deep probability distribution, at a given parameter choice to determine what dimensions of parameter space are sensitive (high magnitude eigenvalue), and which are degenerate (low magnitude eigenvalue) with respect to the emergent property produced. In Figure 1B, the eigenvectors of the Hessian v_1 and v_2 are shown evaluated at the mode of the distribution. The length of the arrows is inversely proportional to the square root of absolute value of their eigenvalues $\lambda_1 = -10.8$ and $\lambda_2 = -2.27$. We quantitatively measured the sensitivity of the model with respect to network syncing along the eigenvectors of the Hessian (Fig. 1B, inset). Sensitivity was measured as the slope coefficient of linear regression fit to network syncing error (the sum of squared differences of each neuron's frequency from 0.53Hz) as a function of parametric perturbation magnitude (maximum 0.25) away from the mode along both orientations indicated by the eigenvector with 100 equally spaced samples. The sensitivity slope coefficient of eigenvector v_1 with respect to network syncing was significant ($\beta = 4.82 \times 10^{-2}$, $p < 10^{-5}$) In contrast, eigenvector v_2 did not identify a dimension of parameter space significantly sensitive to network syncing ($\beta = 8.65 \times 10^{-4}$ with $p = .67$). These sensitivities were compared to all other dimensions of parameter space (100 equally spaced angles from 0 to π), revealing that the Hessian eigenvectors indeed identified the directions of greatest sensitivity and degeneracy (Fig. 1B, inset).

B.2.2 Primary visual cortex

The dynamics of each neural populations average rate $x = [x_E \ x_P \ x_S \ x_V]^\top$ are given by:

$$\tau \frac{dx}{dt} = -x + [Wx + h]_+^n \quad (54)$$

Some neuron-types largely lack synaptic projections to other neuron-types [43], and it is popular to only consider a subset of the effective connectivities [24, 44, 45].

$$W = \begin{bmatrix} W_{EE} & W_{EP} & W_{ES} & 0 \\ W_{PE} & W_{PP} & W_{PS} & 0 \\ W_{SE} & 0 & 0 & W_{SV} \\ W_{VE} & W_{VP} & W_{VS} & 0 \end{bmatrix} \quad (55)$$

By consolidating information from many experimental datasets, Billeh et al. [47] produce estimates

⁸⁴⁶ of the synaptic strength (in mV)

$$M = \begin{bmatrix} 0.36 & 0.48 & 0.31 & 0.28 \\ 1.49 & 0.68 & 0.50 & 0.18 \\ 0.86 & 0.42 & 0.15 & 0.32 \\ 1.31 & 0.41 & 0.52 & 0.37 \end{bmatrix} \quad (56)$$

⁸⁴⁷ and connection probability

$$C = \begin{bmatrix} 0.16 & 0.411 & 0.424 & 0.087 \\ 0.395 & .451 & 0.857 & 0.02 \\ 0.182 & 0.03 & 0.082 & 0.625 \\ 0.105 & 0.22 & 0.77 & 0.028 \end{bmatrix} \quad (57)$$

⁸⁴⁸ Multiplying these connection probabilities and synaptic efficacies gives us an effective connectivity
⁸⁴⁹ matrix:

$$W_{\text{full}} = C \odot M = \begin{bmatrix} 0.16 & 0.411 & 0.424 & 0.087 \\ 0.395 & .451 & 0.857 & 0.02 \\ 0.182 & 0.03 & 0.082 & 0.625 \\ 0.105 & 0.22 & 0.77 & 0.028 \end{bmatrix} \quad (58)$$

⁸⁵⁰ We used the entries of this full effective connectivity matrix that are not considered to be ineffectual
⁸⁵¹ (Equation 55).

⁸⁵² We look at how this four-dimensional nonlinear dynamical model of V1 responds to different inputs,
⁸⁵³ and compare the predictions of the linear response to the approximate posteriors obtained through
⁸⁵⁴ EPI. The input to the system is the sum of a baseline input $b = [1 \ 1 \ 1 \ 1]^T$ and a differential
⁸⁵⁵ input dh :

$$h = b + dh \quad (59)$$

⁸⁵⁶ All simulations of this system had $T = 100$ time points, a time step $dt = 5\text{ms}$, and time constant
⁸⁵⁷ $\tau = 20\text{ms}$. And the system was initialized to a random draw $x(0)_i \sim \mathcal{N}(1, 0.01)$.

⁸⁵⁸ We can describe the dynamics of this system more generally by

$$\dot{x}_i = -x_i + f(u_i) \quad (60)$$

⁸⁵⁹ where the input to each neuron is

$$u_i = \sum_j W_{ij} x_j + h_i \quad (61)$$

860 Let $F_{ij} = \gamma_i \delta(i, j)$, where $\gamma_i = f'(u_i)$. Then, the linear response is

$$\frac{dx_{ss}}{dh} = F(W \frac{dx_{ss}}{dh} + I) \quad (62)$$

861 which is calculable by

$$\frac{dx_{ss}}{dh} = (F^{-1} - W)^{-1} \quad (63)$$

862 This calculation is used to produce the magenta lines in Figure 2C, which show the linearly predicted
863 inputs that generate a response from two standard deviations (of \mathcal{B}) below and above y .

864 The emergent property we considered was the first and second moments of the change in steady
865 state rate dx_{ss} between the baseline input $h = b$ and $h = b + dh$. We use the following notation to
866 indicate that the emergent property statistics were set to the following values:

$$\mathcal{B}(\alpha, y) \triangleq \mathbb{E} \begin{bmatrix} dx_{\alpha,ss} \\ (dx_{\alpha,ss} - y)^2 \end{bmatrix} = \begin{bmatrix} y \\ 0.01^2 \end{bmatrix} \quad (64)$$

867 In the final analysis for this model, we sweep the input one neuron at a time away from the mode
868 of each inferred distributions $dh^* = z^* = \text{argmax}_z \log q_\theta(z \mid \mathcal{B}(\alpha, 0.1))$. The differential responses
869 $\delta x_{\alpha,ss}$ are examined at perturbed inputs $h = b + dh^* + \delta h_\alpha \hat{u}_\alpha$ where \hat{u}_α is a unit vector in the
870 dimension of α and δx is evaluated at 101 equally spaced samples of δh_α from -15 to 15.

871 We measured the linear regression slope between neuron-types of δx and δh to confirm the hy-
872 potheses H1-H3 (H4 is simply observing the nonmonotonicity) and report the p values for tests of
873 non-zero slope.

874 H1: the neuron-type responses are sensitive to their direct inputs. E-population: $\beta = 1.62$,
875 $p = 2.97 \times 10^{-31}$ (Fig. 3A black), P-population: $\beta = 1.06$, $p = 1.64 \times 10^{-34}$ (Fig. 3B
876 blue), S-population: $\beta = 6.80$, $p = 2.65 \times 10^{-26}$ (Fig. 3C red), V-population: $\beta = 6.41$,
877 $p = 1.36 \times 10^{-25}$ (Fig. 3D green).

878 H2: the E-population ($\beta = 0$, $p = 1$) and P-populations ($\beta = 0$, $p = 1$) are not affected by
879 δh_V (Fig. 3A green, 3B green);

880 H3: the S-population is not affected by δh_P ($\beta = 0$, $p = 1$) (Fig. 3C blue);

881

882 For each $\mathcal{B}(\alpha, y)$ with $\alpha \in \{E, P, S, V\}$ and $y \in \{0.1, 0.5\}$, we ran EPI using a real NVP architecture
883 of four masks layers with two hidden layers of 10 units, mapped to a support of $z_i \in [-5, 5]$ with
884 no batch normalization. We used an augmented Lagrangian coefficient of $c_0 = 10^5$, a batch size
885 $n = 1000$, set $\nu = 0.5$. The EPI distributions shown in Fig. 2 are the converged distributions with
886 maximum entropy across random seeds.

887 Here, we demonstrate that the algorithm does not necessarily need to start at such an agnostic
 888 initialization. We set the parameters of the Gaussian initialization μ_{init} and Σ_{init} to the mean and
 889 covariance of random samples $z^{(i)} \sim \mathcal{U}(-5, 5)$ that produced emergent property statistic $dx_{\alpha,ss}$
 890 within a bound ϵ of the emergent property value y . $\epsilon = 0.01$ was set to be one standard deviation
 891 of the emergent property value according to the emergent property value 0.01^2 of the variance
 892 emergent property statistic. This is the only application in the study where such an informed
 893 initialization was used.

894 **B.2.3 Superior colliculus**

895 In the model of Duan et al [25], there are four total units: two in each hemisphere corresponding to
 896 the Pro/Contra and Anti/Ipsi populations. They are denoted as left Pro (LP), left Anti (LA), right
 897 Pro (RP) and right Anti (RA). Each unit has an activity (x_α) and internal variable (u_α) related
 898 by

$$x_\alpha = \left(\frac{1}{2} \tanh \left(\frac{u_\alpha - \epsilon}{\zeta} \right) + \frac{1}{2} \right) \quad (65)$$

899 where $\alpha \in \{LP, LA, RA, RP\}$ $\epsilon = 0.05$ and $\zeta = 0.5$ control the position and shape of the nonlin-
 900 earity, respectively.

901 We order the elements of x and u in the following manner

$$x = \begin{bmatrix} x_{LP} \\ x_{LA} \\ x_{RP} \\ x_{RA} \end{bmatrix} \quad u = \begin{bmatrix} u_{LP} \\ u_{LA} \\ u_{RP} \\ u_{RA} \end{bmatrix} \quad (66)$$

902 The internal variables follow dynamics:

$$\tau \frac{du}{dt} = -u + Wx + h + \sigma dB \quad (67)$$

903 with time constant $\tau = 0.09s$ and Gaussian noise σdB controlled by the magnitude of $\sigma = 1.0$. The
 904 weight matrix has 8 parameters sW_P , sW_A , vW_{PA} , vW_{AP} , hW_P , hW_A , dW_{PA} , and dW_{AP} (Fig.
 905 4B).

$$W = \begin{bmatrix} sW_P & vW_{PA} & hW_P & dW_{PA} \\ vW_{AP} & sW_A & dW_{AP} & hW_A \\ hW_P & dW_{PA} & sW_P & vW_{PA} \\ dW_{AP} & hW_A & vW_{AP} & sW_A \end{bmatrix} \quad (68)$$

906 The system receives five inputs throughout each trial, which has a total length of 1.8s.

$$h = h_{\text{rule}} + h_{\text{choice-period}} + h_{\text{light}} \quad (69)$$

907 There are rule-based inputs depending on the condition,

$$h_{P,\text{rule}}(t) = \begin{cases} I_{P,\text{rule}} \begin{bmatrix} 1 & 0 & 0 & 1 \end{bmatrix}^\top, & \text{if } t \leq 1.2s \\ 0, & \text{otherwise} \end{cases} \quad (70)$$

908

$$h_{A,\text{rule}}(t) = \begin{cases} I_{A,\text{rule}} \begin{bmatrix} 0 & 1 & 1 & 0 \end{bmatrix}^\top, & \text{if } t \leq 1.2s \\ 0, & \text{otherwise} \end{cases} \quad (71)$$

909 a choice-period input,

$$h_{\text{choice}}(t) = \begin{cases} I_{\text{choice}} \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix}^\top, & \text{if } t > 1.2s \\ 0, & \text{otherwise} \end{cases} \quad (72)$$

910 and an input to the right or left-side depending on where the light stimulus is delivered.

$$h_{\text{light}}(t) = \begin{cases} I_{\text{light}} \begin{bmatrix} 1 & 1 & 0 & 0 \end{bmatrix}^\top, & \text{if } t > 1.2s \text{ and Left} \\ I_{\text{light}} \begin{bmatrix} 0 & 0 & 1 & 1 \end{bmatrix}^\top, & \text{if } t > 1.2s \text{ and Right} \\ 0, & t \leq 1.2s \end{cases} \quad (73)$$

911 The input parameterization was fixed to $I_{P,\text{rule}} = 10$, $I_{A,\text{rule}} = 10$, $I_{\text{choice}} = 2$, and $I_{\text{light}} = 1$

912 To produce a Bernoulli rate of p_{LP} in the Left, Pro condition, let \hat{p}_i be the empirical average steady
913 state (ss) response (final x_{LP} at end of task) over $M=500$ Gaussian noise draws for a given SC
914 model parameterization z_i :

$$\hat{p}_i = \mathbb{E}_{\sigma dB} [x_{LP} | s = L, c = P, z = z_i] = \frac{1}{M} \sum_{j=1}^M x_{LP}(s = L, c = P, z = z_i, \sigma dB_j) \quad (74)$$

915 where from here on x_α denotes the steady state activity at the end of the trial. For the first
916 emergent property statistic, the average over EPI samples (from $q_\theta(z)$) is set to the desired value
917 p_{LP} :

$$\mathbb{E}_{z_i \sim q_\phi} [\mathbb{E}_{\sigma dB} [x_{LP,ss} | s = L, c = P, z = z_i]] = \mathbb{E}_{z_i \sim q_\phi} [\hat{p}_i] = p_{LP} \quad (75)$$

918 For the next emergent property statistic, we ask that the variance of the steady state responses
919 across Gaussian draws, is the Bernoulli variance for the empirical rate \hat{p}_i .

$$\mathbb{E}_{z \sim q_\phi} [\sigma_{err}^2] = 0 \quad (76)$$

920

$$\sigma_{err}^2 = \text{Var}_{\sigma dB} [x_{LP} | s = L, c = P, z = z_i] - \hat{p}_i(1 - \hat{p}_i) \quad (77)$$

921 We have an additional constraint that the Pro neuron on the opposite hemisphere should have the
 922 opposite value (0 and 1). We can enforce this with a final constraint:

$$\mathbb{E}_{z \sim q_\phi} [d_P] = \mathbb{E}_{\sigma dB} [(x_{LP} - x_{RP})^2 | s = L, c = P, z = z_i] = 1 \quad (78)$$

923 Since the maximum variance of a random variable bounded from 0 to 1 is the Bernoulli variance
 924 $\hat{p}(1 - \hat{p})$, and the maximum squared difference between two variables bounded from 0 to 1 is 1, we
 925 do not need to control the second moment of these test statistics. In practice, these variables are
 926 dynamical system states and can only exponentially decay (or saturate) to 0 (or 1), so the Bernoulli
 927 variance error and squared difference constraints can only be undershot. This is important to be
 928 mindful of when evaluating the convergence criteria. Instead of using our usual hypothesis testing
 929 criteria for convergence to the emergent property, we set a slack variable threshold only for these
 930 technically infeasible emergent property values to 0.05.

931 Training DSNs to learn distributions of dynamical system parameterizations that produce Bernoulli
 932 responses at a given rate (with small variance around that rate) was harder to do than expected.
 933 There is a pathology in this optimization setup, where the learned distribution of weights is bimodal
 934 attributing a fraction p of the samples to an expansive mode (which always sends x_{LP} to 1), and a
 935 fraction $1 - p$ to a decaying mode (which always sends x_{LP} to 0). This pathology was avoided using
 936 an inequality constraint prohibiting parameter samples that resulted in low variance of responses
 937 across noise.

938 In total, the emergent property of rapid task switching at accuracy level p was defined as

$$\mathcal{B}(p) \triangleq \mathbb{E} \begin{bmatrix} \hat{p}_P \\ \hat{p}_A \\ (\hat{p}_P - p)^2 \\ (\hat{p}_A - p)^2 \\ \sigma_{P,err}^2 \\ \sigma_{A,err}^2 \\ d_P \\ d_A \end{bmatrix} = \begin{bmatrix} p \\ p \\ 0.15^2 \\ 0.15^2 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix} \quad (79)$$

939 For each accuracy level p , we ran EPI for 10 different random seeds using an architecture of 10
 940 planar flows with a support of $z \in \mathbb{R}^8$. We used an augmented Lagrangian coefficient of $c_0 = 10^2$, a

batch size $n = 300$, and set $\nu = 0.5$, and initialized $q_\theta(z)$ to produce an isotropic Gaussian of zero mean with standard deviation $\sigma_{\text{init}} = 1$. The EPI distributions shown in Fig. 4 are the converged distributions with maximum entropy across random seeds.

We report significant correlations r and their pvalues from Figure 4E. Correlations were measured from 5,000 samples of $q_\theta(z | \mathcal{B}(p))$ and p-values are reported for one-tailed tests, since we predict positive correlation between task performance and λ_{task} , and negative correlation between task performance in and λ_{side} .

λ	\hat{p}	$q_\theta(z)$	r	p-value
λ_{task}	\hat{p}_P	$q(z \mathcal{B}(60\%))$	1.24×10^{-01}	3.04×10^{-18}
λ_{task}	\hat{p}_P	$q(z \mathcal{B}(70\%))$	7.56×10^{-01}	0.00
λ_{task}	\hat{p}_P	$q(z \mathcal{B}(80\%))$	4.59×10^{-01}	2.76×10^{-259}
λ_{task}	\hat{p}_P	$q(z \mathcal{B}(90\%))$	3.76×10^{-01}	1.83×10^{-167}
λ_{task}	\hat{p}_A	$q(z \mathcal{B}(60\%))$	4.80×10^{-02}	1.38×10^{-03}
λ_{task}	\hat{p}_A	$q(z \mathcal{B}(70\%))$	2.08×10^{-01}	7.17×10^{-50}
λ_{task}	\hat{p}_A	$q(z \mathcal{B}(80\%))$	4.84×10^{-01}	3.45×10^{-291}
λ_{task}	\hat{p}_A	$q(z \mathcal{B}(90\%))$	4.25×10^{-01}	1.04×10^{-217}
λ_{side}	\hat{p}_P	$q(z \mathcal{B}(50\%))$	-7.57×10^{-02}	1.69×10^{-07}
λ_{side}	\hat{p}_P	$q(z \mathcal{B}(60\%))$	-6.73×10^{-02}	3.87×10^{-06}
λ_{side}	\hat{p}_P	$q(z \mathcal{B}(70\%))$	-4.86×10^{-01}	4.43×10^{-295}
λ_{side}	\hat{p}_P	$q(z \mathcal{B}(80\%))$	-1.43×10^{-01}	5.97×10^{-24}
λ_{side}	\hat{p}_P	$q(z \mathcal{B}(90\%))$	-1.93×10^{-01}	8.08×10^{-43}
λ_{side}	\hat{p}_A	$q(z \mathcal{B}(50\%))$	-1.33×10^{-02}	6.94×10^{-01}
λ_{side}	\hat{p}_A	$q(z \mathcal{B}(60\%))$	-7.60×10^{-02}	1.47×10^{-07}
λ_{side}	\hat{p}_A	$q(z \mathcal{B}(70\%))$	-2.73×10^{-01}	5.23×10^{-86}
λ_{side}	\hat{p}_A	$q(z \mathcal{B}(80\%))$	-2.74×10^{-01}	1.30×10^{-86}
λ_{side}	\hat{p}_A	$q(z \mathcal{B}(90\%))$	-1.82×10^{-02}	3.95×10^{-01}

Table 1: Table of significant correlation values from Fig. 4E.

948 B.2.4 Rank-1 RNN

Recent work establishes a link between RNN connectivity weights and the resulting dynamical responses of the network, using dynamic mean field theory (DMFT) [26]. Specifically, DMFT

describes the properties of activity in infinite-size neural networks given a distribution on the connectivity weights. In such a model, the connectivity of a rank-1 RNN (which was sufficient for the Gaussian posterior conditioning task), has weight matrix W , which is the sum of a random component with strength determined by g and a structured component determined by the outer product of vectors m and n :

$$W = g\chi + \frac{1}{N}mn^\top, \quad (80)$$

where $\chi_{ij} \sim \mathcal{N}(0, \frac{1}{N})$, and the entries of m and n are drawn from Gaussian distributions $m_i \sim \mathcal{N}(M_m, 1)$ and $n_i \sim \mathcal{N}(M_n, 1)$. From such a parameterization, this theory produces consistency equations for the dynamic mean field variables in terms of parameters like g , M_m , and M_n , which we study in Section 3.5. That is the dynamic mean field variables (e.g. the activity along a vector κ_v , the total variance Δ_0 , structured variance Δ_∞ , and the chaotic variance Δ_T) are written as functions of one another in terms of connectivity parameters. The values of these variables can be used obtained using a nonlinear system of equations solver. These dynamic mean field variables are then cast as task-relevant variables with respect to the context of the provided inputs. Mastrogiovise et al. designed low-rank RNN connectivities via minimalist connectivity parameters to solve canonical tasks from behavioral neuroscience.

We consider the DMFT equation solver as a black box that takes in a low-rank parameterization z (e.g. $z = [g \ M_m \ M_n]$) and outputs the values of the dynamic mean field variables, of which we cast κ_r and Δ_T as task-relevant variables μ_{post} and σ_{post}^2 in the Gaussian posterior conditioning toy example. Importantly, the solution produced by the solver is differentiable with respect to the input parameters, allowing us to use DMFT to calculate the emergent property statistics in EPI to learn distributions on such connectivity parameters of RNNs that execute tasks.

Specifically, we solve for the mean field variables κ_r , κ_n , Δ_0 and Δ_∞ , where the readout is nominally chosen to point in the unit orthant $r = [1 \ \dots \ 1]^\top$. The consistency equations for these variables in the presence of a constant input $h = y - (n - M_n)$ can be derived following [26] are

$$\begin{aligned} \kappa_r &= G_1(\kappa_r, \kappa_n, \Delta_0, \Delta_\infty) = M_m \kappa_n + y \\ \kappa_n &= G_2(\kappa_r, \kappa_n, \Delta_0, \Delta_\infty) = M_n \langle [\phi_i] \rangle + \langle [\phi'_i] \rangle \\ \frac{\Delta_0^2 - \Delta_\infty^2}{2} &= G_3(\kappa_r, \kappa_n, \Delta_0, \Delta_\infty) = g^2 \left(\int \mathcal{D}z \Phi^2(\kappa_r + \sqrt{\Delta_0} z) - \int \mathcal{D}z \int \mathcal{D}x \Phi(\kappa_r + \sqrt{\Delta_0 - \Delta_\infty} x + \sqrt{\Delta_\infty} z) \right) \\ &\quad + (\kappa_n^2 + 1)(\Delta_0 - \Delta_\infty) \\ \Delta_\infty &= G_4(\kappa_r, \kappa_n, \Delta_0, \Delta_\infty) = g^2 \int \mathcal{D}z \left[\int \mathcal{D}x \phi(\kappa_r + \sqrt{\Delta_0 - \Delta_\infty} x + \sqrt{\Delta_\infty} z) \right]^2 + \kappa_n^2 + 1 \end{aligned} \quad (81)$$

975 where here z is a gaussian integration variable. We can solve these equations by simulating the
 976 following Langevin dynamical system to a steady state.

$$\begin{aligned}
 l(t) &= \frac{\Delta_0(t)^2 - \Delta_\infty(t)^2}{2} \\
 \Delta_0(t) &= \sqrt{2x(t) + \Delta_\infty(t)^2} \\
 \frac{d\kappa_r(t)}{dt} &= -\kappa_r(t) + F(\kappa_r(t), \kappa_n(t), \Delta_0(t), \Delta_\infty(t)) \\
 \frac{d\kappa_n(t)}{dt} &= -\kappa_n + G(\kappa_r(t), \kappa_n(t), \Delta_0(t), \Delta_\infty(t)) \\
 \frac{dl(t)}{dt} &= -l(t) + H(\kappa_r(t), \kappa_n(t), \Delta_0(t), \Delta_\infty(t)) \\
 \frac{d\Delta_\infty(t)}{dt} &= -\Delta_\infty(t) + L(\kappa_r(t), \kappa_n(t), \Delta_0(t), \Delta_\infty(t))
 \end{aligned} \tag{82}$$

977 Then, the chaotic variance, which is necessary for the Gaussian posterior conditioning example, is
 978 simply calculated via

$$\Delta_T = \Delta_0 - \Delta_\infty \tag{83}$$

979 We ran EPI using a real NVP architecture of two masks and two layers per mask with 10 units
 980 mapped to a support of $z \in \left[\begin{bmatrix} 0 & -5 & -5 \end{bmatrix}, \begin{bmatrix} 5 & 5 & 5 \end{bmatrix} \right]$ with no batch normalization. We used
 981 an augmented Lagrangian coefficient of $c_0 = 1$, a batch size $n = 300$, set $\nu = 0.2$, and initialized
 982 $q_\theta(z)$ to produce an isotropic Gaussian with mean $\mu_{\text{init}} = \begin{bmatrix} 2.5 & 0 & 0 \end{bmatrix}$ with standard deviation
 983 $\sigma_{\text{init}} = 2.0$. The EPI distributions shown in Fig. 4 are the converged distributions with maximum
 984 entropy across five random seeds.

985 In addition to the Gaussian posterior conditioning example in Section 3.5, we modeled two tasks
 986 from Mastrogiosse et al.: noisy detection and context-dependent discrimination. We used the
 987 same theoretical equations and task setups described in their study.

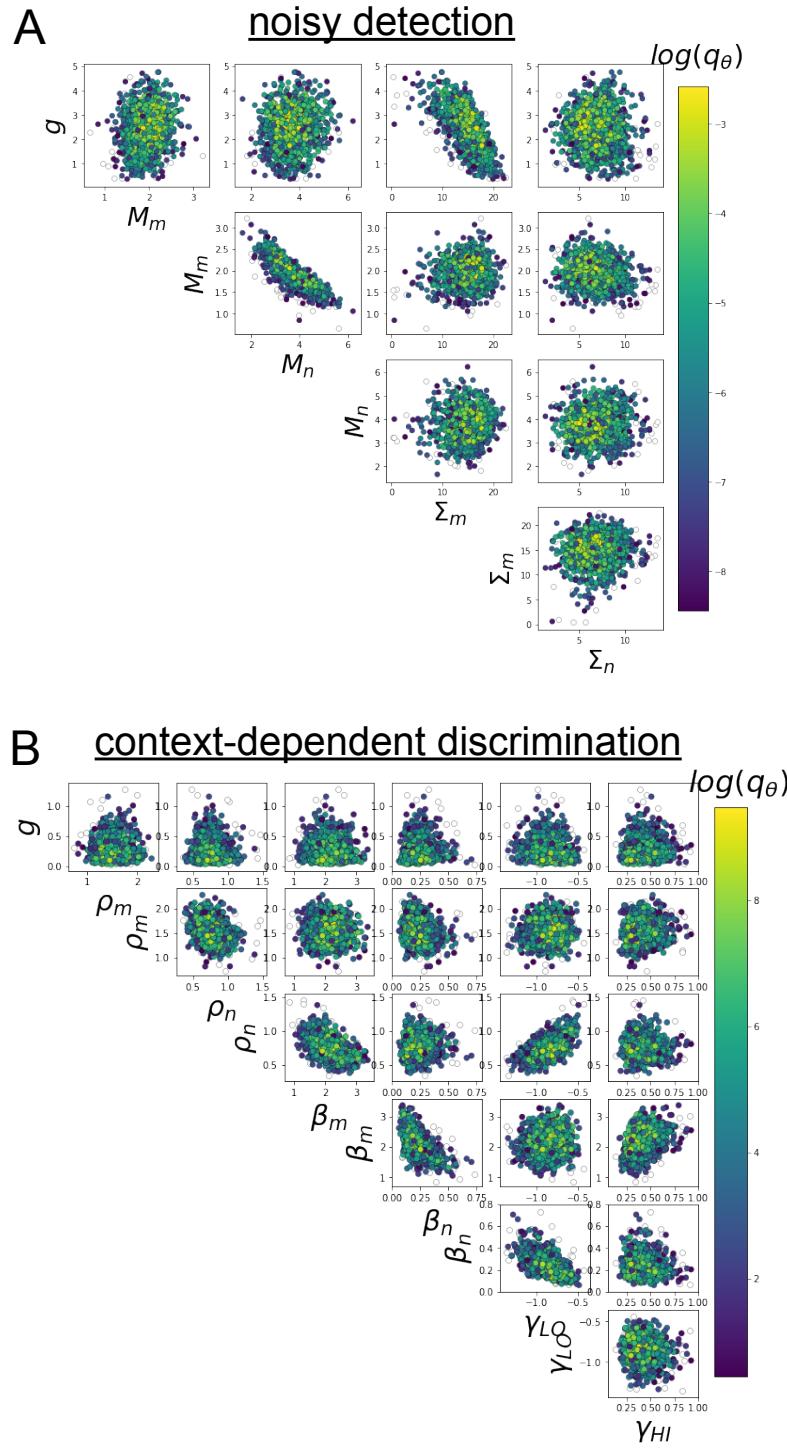


Fig. S5: A. EPI for rank-1 networks doing noisy discrimination. B. EPI for rank-2 networks doing context-dependent discrimination. See [26] for theoretical equations and task description.