



Available online at [www.sciencedirect.com](http://www.sciencedirect.com)



Journal of Physiology - Paris xxx (2004) xxx-xxx

Journal of  
**Physiology**  
Paris

[www.elsevier.com/locate/jphysparis](http://www.elsevier.com/locate/jphysparis)

## Optimal computation with attractor networks

Peter E. Latham <sup>a,\*</sup>, Sophie Deneve <sup>b</sup>, Alexandre Pouget <sup>c</sup>

<sup>a</sup> Department of Neurobiology, University of California at Los Angeles, Los Angeles, CA 90095-1763, USA

<sup>b</sup> Gatsby Computational Neuroscience Unit, University College London, London WC1 3AR, UK

<sup>c</sup> Department of Brain and Cognitive Sciences, University of Rochester, Rochester, NY 14627, USA

### Abstract

We investigate the ability of multi-dimensional attractor networks to perform reliable computations with noisy population codes. We show that such networks can perform computations as reliably as possible—meaning they can reach the Cramér-Rao bound—so long as the noise is small enough. “Small enough” depends on the properties of the noise, especially its correlational structure. For many correlational structures, noise in the range of what is observed in the cortex is sufficiently small that biologically plausible networks can compute optimally. We demonstrate that this result applies to computations that involve cues of varying reliability, such as the position of an object on the retina in bright versus dim light.

© 2004 Published by Elsevier Ltd.

### 1. Introduction

Many variables in the brain are encoded in the activity of large populations of neurons with bell-shaped tuning curves (see Fig. 1a). A critical question in neuroscience is: how do networks compute with these codes? How can a network extract, for example, the position of an object in head-centered coordinates from the position of the object on the retina and the position of the eyes in the head, given that all three variables are encoded by population activity? Tasks like this are made especially difficult by the variability in neuronal responses (Fig. 1b): neurons never fire with exactly the same pattern twice, even when an animal is performing identical tasks—say responding to the same stimulus, or producing the same motor response [9,14,17].

The fact that population codes are noisy means that information is lost at every stage of processing, so there is pressure to perform computations reliably. To understand the limits of reliability, we consider a scenario in which a network receives as input information about a set of variables, each encoded in population activity, and the network performs some computation based on that input (such as the one mentioned above). The question we ask is: how reliably can the network do this? In other words, how much of the information in

the input can the network extract while it is carrying out the computation?

As a first step toward answering this question, we consider a restricted class of networks for which smooth hills of activity are stable. When a network within this class is initialized with noisy population activity—with noisy hills of activity—the network eventually evolves onto smooth hills, like the one shown in Fig. 1c. Once the smooth hill is obtained, its peak constitutes an estimate of the value of the variable encoded in the noisy hill. The crucial question is whether this estimate can be optimal, that is, whether the estimate can be computed with no loss of information.

For the simple case of a single variable encoded in population activity, as in Fig. 1, we found in our previous work that there is a network that produces an optimal estimate of the encoded variable [13]. This result holds so long as the noise is Poisson and is independent among neurons. In a subsequent study, we extended this finding to networks encoding multiple independent variables [6]. More recently, we presented simulations suggesting an even more general result: networks encoding multiple variables, related to one another through nonlinear transformations, can be tuned to perform optimal computation even when the reliability of the input variables change from trial to trial [7].

In this paper we prove the above general result. Our proof applies to the case in which the evolution of the

\* Corresponding author.

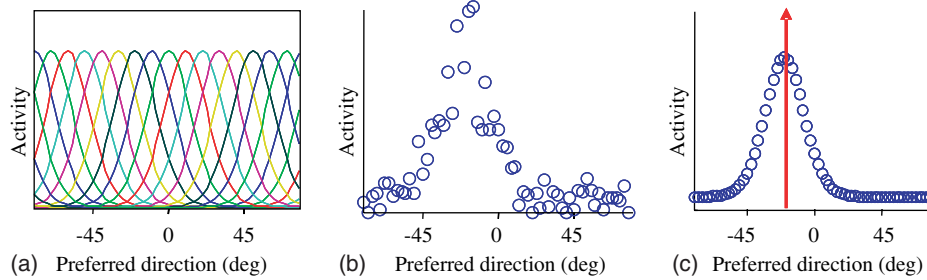


Fig. 1. (a) A set of tuning curves for the direction of motion of an object. Such tuning curves are found throughout the visual cortex, and in particular in area MT. (b) A noisy pattern of activity for a population of neurons. This pattern arose from an object moving at  $-30^\circ$ . (c) A smooth hill of activity. The networks we consider here evolve to a smooth hill like this one when initialized with the noisy pattern in panel b. The position of the peak of the smooth hill (marked with a red arrow) provides an estimate of direction of motion. With proper tuning of its parameters, a network can recover the optimal estimate; that is, it can evolve to a smooth hill without losing any of the information coded in the noisy hill.

69 network is noise-free, which means the only source of  
70 noise is the noise corrupting the input hills. Given this  
71 assumption, we derive conditions for the existence of a  
72 network that can perform optimal computations; that is,  
73 for a network that can manipulate population codes  
74 without losing any of the information in the input. The  
75 conditions are very general and relatively simple; they  
76 depend only on the correlational structure of the noise  
77 in the input. Interestingly, for small enough noise there  
78 is always a network that can perform computations  
79 optimally. However, the size of “small enough” depends  
80 in detail on the correlational structure.

81 Letting the network evolve noise-free is a big  
82 approximation; we make it because it allows us to derive  
83 powerful results telling us when a network can carry out  
84 computations reliably and when it cannot. The more  
85 realistic case of internal noise (synaptic failures, sto-  
86 chastic ion channels, etc.) can be handled by considering  
87 the evolution of probability distributions over neuronal  
88 activity rather than the neuronal activity itself. This case  
89 will be considered in future work; our underlying  
90 assumption in this paper is that, for small enough  
91 internal noise, the deterministic evolution should pro-  
92 vide a reasonable first approximation to the true prob-  
93 abilistic evolution (see Fig. 3).

94 This paper is arranged as follows. In Section 2 we  
95 provide an intuitive explanation of how neuronal net-  
96 works perform efficient estimation. Section 3 contains a  
97 formal derivation of our main result, that *any* recurrent  
98 network exhibiting an  $M$ -dimensional attractor is  
99 capable of performing as well as the best possible esti-  
100 mator in the limit of small noise. We provide an estimate  
101 of the size of the noise for this result to hold, and show  
102 that for uncorrelated noise, or correlated noise that is  
103 stimulus independent, it need only be  $\mathcal{O}(1)$ . However, if  
104 the noise is correlated and stimulus-dependent, it must  
105 be  $\mathcal{O}(1/N)$ . In Section 4 we extend this result to net-  
106 works in which the reliability of stimuli is variable. In  
107 Section 5 we consider an example: correlated, Poisson-  
108 like neurons, for which  $\mathcal{O}(1/N)$  noise is required to

perform optimal computations for the class of networks  
considered in Sections 2 and 3. We show that a network  
does exist that computes optimally for  $\mathcal{O}(1)$  noise.  
However, that network is not so easily implemented in a  
biological network, and does not readily generalize.  
Section 6 contains our summary and conclusions.

## 2. Extracting information from noisy neurons: general considerations

Biological organisms must estimate stimuli from  
noisy neuronal responses. They must be able, for  
example, to translate from the noisy hill of activity in  
Fig. 1a to the value of the variable encoded by that hill,  
or perform a computation by combining the noisy hills  
associated with several variables. (We are using “stimu-  
lus” in a very general sense; a stimulus could be an  
external variable, such as the direction of a moving  
object, or an internal variable, such as the position of  
the eyes relative to the head. Stimuli could even consist  
of some combination of external and internal variables.)  
The question we ask in this paper is: how well can  
biologically plausible networks carry out these estima-  
tion tasks? In particular, can they do as well as the best  
possible estimator; that is, can they reach the Cramér-  
Rao bound [5]? Surprisingly, the answer to the latter  
question is yes, so long as certain conditions are met. In  
the next section we derive those conditions; in this sec-  
tion we provide an intuitive explanation of why bio-  
logically plausible networks might be able to act as  
optimal estimators.

Formally, the brain performs estimation by imple-  
menting a mapping from a set of neuronal responses,  
denoted  $\mathbf{a} \equiv (\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_N)$ , to a stimulus or set of  
stimuli, denoted  $\mathbf{s}$ . For simplicity, in this section we take  
both the stimulus and each of the neuronal responses to  
be one-dimensional; so we let  $(\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_N) \rightarrow$   
 $(a_1, a_2, \dots, a_N)$  and  $\mathbf{s} \rightarrow s$ , where the  $a_i$  and  $s$   
are scalar variables. For example, the  $a_i$  might be firing rates and  $s$

109  
110  
111  
112  
113  
114

115  
116

117  
118  
119  
120  
121  
122  
123  
124  
125  
126  
127  
128  
129  
130  
131  
132  
133  
134  
135  
136  
137

138  
139  
140  
141  
142  
143  
144  
145

146 the direction of a moving object, as in Fig. 1. We show  
147 in the next section, however, that our results apply even  
148 when both the stimulus and the response of each neuron  
149 is multi-dimensional.

150 We start by assuming that some estimator exists; i.e.,  
151 that there is some function of  $\mathbf{a}$ , denoted  $\hat{s}(\mathbf{a})$ , that  
152 provides an estimate of the stimulus,  $s$ . The estimator  
153  $\hat{s}(\mathbf{a})$  can be thought of as a many-to-one mapping from  $\mathbf{a}$   
154 to  $\hat{s}$ . To make this explicit, we write

$$\hat{s} = \hat{s}(\mathbf{a}). \quad (1)$$

156 The observation that allows us to construct a network  
157 estimator out of the general estimator,  $\hat{s}(\mathbf{a})$ , is that Eq.  
158 (1) can be inverted to provide a one-to-many map from  $\hat{s}$   
159 to  $\mathbf{a}$ . Specifically, if we view activity space as an  
160  $N$ -dimensional space whose coordinates are  
161  $(a_1, a_2, \dots, a_N)$ , then, for each value of  $\hat{s}$ , the set of  $a_i$   
162 that satisfies Eq. (1) forms an  $(N - 1)$ -dimensional  
163 subspace, denoted  $\mathbf{a}(\hat{s})$ . The key feature of this subspace  
164 is that every point in it leads to the same estimate,  $\hat{s}$ , of  
165 the stimulus,  $s$ ; i.e., every point in the subspace  $\mathbf{a}(\hat{s})$   
166 produces the same value for  $\hat{s}(\mathbf{a})$ . If we could construct a  
167 network that maps the whole  $(N - 1)$ -dimensional space  
168 to a single point, the location of that point would provide  
169 a natural estimate of  $\hat{s}$ .

170 Attractor networks [4,8,10–12,18,20] could perform  
171 such a mapping. These networks evolve in time, starting  
172 from some initial condition, to an attractor—a sub-  
173 manifold of their full activity space. An attractor net-  
174 work could, then, take as initial conditions the popula-  
175 tion activity,  $\mathbf{a}$ , and evolve in time such that the whole  
176  $(N - 1)$ -dimensional subspace,  $\mathbf{a}(\hat{s})$ , goes eventually  
177 to the same point. In the full  $N$ -dimensional activity space,  
178 such a network would admit a line-attractor, so con-  
179 structing a network estimator out of the general esti-  
180 mator  $\hat{s}(\mathbf{a})$  reduces to the problem of finding the  
181 appropriate line-attractor network.

182 Fig. 2 shows schematically how a line-attractor net-  
183 work could act as an estimator. The activity,  $\mathbf{a}$ , in re-  
184 sponse to a stimulus,  $s$ , corresponds to an initial  
185 condition for the attractor network. Each initial condi-  
186 tion lies on *some*  $(N - 1)$ -dimensional subspace; i.e.,  
187 every  $\mathbf{a}$  solves Eq. (1) for some  $\hat{s}$ . Two such subspaces  
188 are shown in Fig. 2. Under the action of the line-  
189 attractor network, every point in a particular subspace  
190 evolves to the same final point, and that point lies on the  
191 line labeled  $\mathbf{g}(s)$ . For example all points lying on the  
192 sheet  $\mathbf{a}(\hat{s}_1)$  evolve to  $\mathbf{g}(\hat{s}_1)$  and all points on the sheet  
193  $\mathbf{a}(\hat{s}_2)$  evolve to  $\mathbf{g}(\hat{s}_2)$ . The final position on the line  $\mathbf{g}(s)$   
194 represents the network estimate of the stimulus,  $s$ .

195 This analysis indicates that every line-attractor cor-  
196 responds to *some* estimator. The question of interest is:  
197 can a line-attractor network do as well as the best pos-  
198 sible estimator? This is a hard question to answer in  
199 general. However, it is tractable in the limit of small  
200 noise. In this limit, the initial condition,  $\mathbf{a}$ , is near the

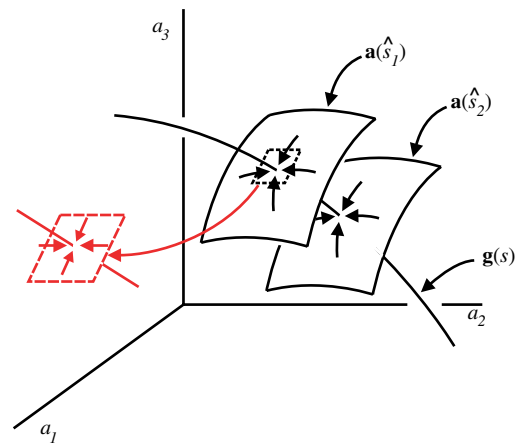


Fig. 2. Schematic (three-dimensional cut) of a line-attractor network that mimics  $\hat{s}(\mathbf{a})$ . The sheets represent  $(N - 1)$ -dimensional subspaces for two different values of  $\hat{s}$ ; points on the sheets satisfy Eq. (1), with  $\hat{s} = \hat{s}_1$  for the upper one and  $\hat{s} = \hat{s}_2$  for the lower. Arrows on the sheets indicate trajectories. The line labeled  $\mathbf{g}(s)$  is the line-attractor: all initial conditions evolve to some point on this line. The position on the line provides the estimate,  $\hat{s}$ , of the true stimulus,  $s$ . A blowup of the region near the line-attractor, shown in red, indicates that the subspaces are locally flat. Consequently, for initial conditions close enough to the line-attractor,  $\mathbf{g}(s)$ , the trajectories of the line-attractor network are well approximated by straight lines. For a network to mimic the estimator  $\hat{s}(\mathbf{a})$ , at least in the small noise limit, it is necessary that the trajectories be parallel to  $\mathbf{a}(\hat{s})$  when  $\mathbf{a}$  is near  $\mathbf{g}(s)$ .

201 line-attractor,  $\mathbf{g}(s)$ , which allows us to treat the  $(N - 1)$ -  
202 dimensional subspaces as linear spaces and the trajec-  
203 tories as straight and locally parallel to  $\mathbf{a}(\hat{s})$  (red blowup  
204 in Fig. 2). Consequently, we can use linear analysis to  
205 compute the quality of the network estimator for any  
206 line-attractor network—that is, we can compute how  
207 well  $\hat{s}$  approximates  $s$ . This is a key point, because  
208 knowing the quality of the estimator for any line-  
209 attractor network allows us to find the best possible line-  
210 attractor network. Moreover, we can show that the best  
211 possible line-attractor network really is good: if the  
212 noise is small— $\mathbf{a}(\hat{s})$  is close to the line  $\mathbf{g}(s)$ , where close  
213 is relative to the curvature of  $\mathbf{a}(\hat{s})$ —we are guaranteed  
214 that the best possible line-attractor network does as well  
215 as the optimal estimator, the latter assessed by the  
216 Cramér-Rao bound.

217 The quality of the linear approximation depends, of  
218 course, on the smoothness of  $\mathbf{a}(\hat{s})$ : if  $\mathbf{a}(\hat{s})$  exhibits sharp  
219 curvature, then our analysis applies only if the noise is  
220 very small (see Section 5 for an example). In the extreme  
221 case in which  $\mathbf{a}(\hat{s})$  exhibits one or more singularities, our  
222 analysis would break down if the line-attractor passed  
223 through any of them. Thus, the smoothness of  $\mathbf{a}(\hat{s})$  must  
224 be checked on a case-by-case basis. For the remainder of  
225 this paper, however, we simply assume that  $\mathbf{a}(\hat{s})$  is lo-  
226 cally smooth.

227 Although the above discussion focused on a one-  
228 dimensional stimulus and one-dimensional responses,

229 the ideas apply to higher-dimensional stimuli and re- 278  
230 sponses as well. In particular they apply to cases where 279  
231 several population codes are combined, as in the three- 280  
232 dimensional case alluded to in the introduction (popu- 281  
233 lations codes for the position of an object on the retina 282  
234 and the position of the eyes in the head are combined to  
235 produce a population code for the position of the object  
236 relative to the head).

### 237 3. Constructing networks that perform optimal estimation

238 We now show explicitly that attractor networks can  
239 act as optimal estimators, in the sense that the network  
240 estimate of a set of stimuli from noisy neuronal re-  
241 sponses is as good as the best possible estimator. We do  
242 this in three steps: we (1) analyze the linearized dynamics  
243 of an attractor network, (2) derive an expression for the  
244 performance of the network in terms of the distance  
245 between the network estimates and the true stimuli, and  
246 (3) show how network parameters can be modified to  
247 optimize the estimates.

248 The problem we consider is the following. A set of  $M$   
249 stimuli produce a particular pattern of activity. That  
250 pattern of activity is fed into a recurrent network that  
251 supports an  $M$ -dimensional attractor. The network then  
252 evolves deterministically in time until it converges onto  
253 the attractor (more accurately, until it get exponentially  
254 close to the attractor). The point on the attractor it  
255 converges to represents the network estimate of the  $M$   
256 stimuli. If we denote the stimuli as  $\mathbf{s} = (s_1, s_2, \dots, s_M)$   
257 and the estimates as  $\hat{\mathbf{s}} = (\hat{s}_1, \hat{s}_2, \dots, \hat{s}_M)$ , we can think of  
258 this process as the mapping  $\mathbf{s} \rightarrow \mathbf{a}(0) \rightarrow \hat{\mathbf{s}}$ , where the  
259 mapping from the stimulus to the initial activity,  $\mathbf{a}(0)$ , is  
260 probabilistic and the mapping from the initial activity to  
261 the stimulus estimate is deterministic.

262 A key observation is that the second half of the  
263 mapping, from  $\mathbf{a}(0)$  to  $\hat{\mathbf{s}}$ , may be implemented with an  
264 attractor network that evolves deterministically in time  
265 according to the equation

$$\tau \frac{d\mathbf{a}(t)}{dt} = \mathbf{H}(\mathbf{a}(t)) - \mathbf{a}(t), \quad (2)$$

267 where  $\tau$  is a time constant,  $\mathbf{a}(t) = (\mathbf{a}_1(t), \mathbf{a}_2(t), \dots, \mathbf{a}_N(t))$   
268 represents the activity of the  $N$  neurons in the network,  
269 and  $\mathbf{H}(\mathbf{a})$  is a function that contains all the details about  
270 the network—its connectivity and single neuron and  
271 synaptic properties. (The  $\mathbf{a}_i(t)$  are vectors because the  
272 response of a single neuron may be multi-dimensional—  
273 latency to the first spike and spike count, for example.)  
274 Our underlying assumption is that  $\mathbf{H}(\mathbf{a})$  is such that the  
275 network admits an  $M$ -dimensional attractor; that is,  
276 there is some smooth function,  $\mathbf{g}(\mathbf{s})$ , satisfying

$$\mathbf{g}(\mathbf{s}) = \mathbf{H}(\mathbf{g}(\mathbf{s})). \quad (3)$$

Since  $\mathbf{s}$  is an  $M$ -component vector,  $\mathbf{g}(\mathbf{s})$  is an  $M$ -dimen- 278  
sional manifold. 279

The network is initialized by transient input at time 280  
 $t = 0$ ; this input has both a deterministic and noise 281  
component, 282

$$\mathbf{a}(0) = \mathbf{f}(\mathbf{s}) + \mathbf{N}(\mathbf{s}). \quad (4)$$

Here  $\mathbf{f}(\mathbf{s})$  is the deterministic tuning curve and  $\mathbf{N}(\mathbf{s})$  is 284  
the noise. In the limit  $t \rightarrow \infty$ ,  $\mathbf{a}(t)$  approaches the 285  
attractor; i.e.,  $\lim_{t \rightarrow \infty} \mathbf{a}(t) = \mathbf{g}(\hat{\mathbf{s}})$ . The point on the 286  
attractor,  $\hat{\mathbf{s}}$ , is the network estimate of the stimuli,  $\mathbf{s}$ . 287

Because the initial conditions are generated proba- 288  
bilistically, the estimate will be different on each trial. 289  
We will assume here that the network is unbiased; that 290  
is, averaged over trials,  $\hat{\mathbf{s}}$  is equal to  $\mathbf{s}$ . Thus, the quality 291  
of the network is determined by how close  $\hat{\mathbf{s}}$  is to  $\mathbf{s}$  on 292  
average. For close, we will use the determinant of the 293  
covariance matrix. The covariance matrix, denoted 294  
 $\langle \delta s_k \delta s_l \rangle$ , is given by 295

$$\langle \delta s_k \delta s_l \rangle = \langle (\hat{s}_k(\mathbf{a}) - s_k)(\hat{s}_l(\mathbf{a}) - s_l) \rangle.$$

This expression is, of course, only valid for unbiased 297  
estimators. We use the log of the covariance matrix to 298  
asses the quality of the estimator because it determines, 299  
to a large extent, the mutual information between the 300  
noisy neuronal responses,  $\mathbf{a}$ , and the stimulus,  $\mathbf{s}$ : the 301  
smaller the determinant of the covariance matrix, the 302  
larger the mutual information [3]. (Strictly speaking, this 303  
result applies only when the neurons are uncorrelated. 304  
We believe it applies also to correlated neurons, as long 305  
as the determinant of the covariance matrix is small. In 306  
any case, it is a good starting point.) 307

To compute the covariance matrix, we take a per- 308  
turbative approach: we linearize Eq. (2) around a point 309  
on the attractor, compute the trajectories analytically, 310  
and find the approximate final position on the attractor 311  
given the initial condition. A difficulty arises because, 312  
unlike point (0-dimensional) attractors, there is not any 313  
unique point on the  $M$ -dimensional attractor to linearize 314  
around. Because of this nonuniqueness, for now we 315  
perturb around an arbitrary point, say  $\mathbf{a} = \mathbf{g}(\tilde{\mathbf{s}})$ . In 316  
principle it does not matter what we choose for  $\tilde{\mathbf{s}}$ , so 317  
long as it is close to the starting point,  $\mathbf{a}(0)$ . However, as 318  
we will see below, there is one especially convenient 319  
choice for  $\tilde{\mathbf{s}}$ . 320

Letting 321

$$\mathbf{a}(t) = \mathbf{g}(\tilde{\mathbf{s}}) + \delta \mathbf{a}(t), \quad (5)$$

inserting Eq. (5) into Eq. (2) and keeping only linear 323  
terms, we find that  $\delta \mathbf{a}(t)$  evolves according to 324

$$\frac{d\delta \mathbf{a}}{dt} = \mathbf{J}(\tilde{\mathbf{s}}) \cdot \delta \mathbf{a}, \quad (6)$$

where  $\mathbf{J}$  is the Jacobian evaluated on the attractor, 326

$$J_{ij}(\tilde{\mathbf{s}}) \equiv \frac{\partial H_i(\mathbf{g}(\tilde{\mathbf{s}}))}{\partial g_j(\tilde{\mathbf{s}})} - \delta_{ij}, \quad (7)$$

328  $\delta_{ij}$  is the Kronecker delta, and we are using standard  
329 dot-product notation: the  $i$ th component of  $\mathbf{J} \cdot \delta\mathbf{a}$  is  
330  $\sum_j J_{ij} \delta a_j$ .  
331 Eq. (6) has the solution

$$\delta\mathbf{a}(t) = \exp(\mathbf{J}(\tilde{\mathbf{s}})t) \cdot \delta\mathbf{a}(0). \quad (8)$$

333 To cast Eq. (8) in a more useful form, we re-express  $\mathbf{J}$   
334 using of its eigenvector expansion,

$$\mathbf{J}(\tilde{\mathbf{s}}) = \sum_k \lambda_k(\tilde{\mathbf{s}}) \mathbf{v}_k(\tilde{\mathbf{s}}) \mathbf{v}_k^\dagger(\tilde{\mathbf{s}}),$$

336 where  $\mathbf{v}_k(\tilde{\mathbf{s}})$  is the eigenvector of  $\mathbf{J}(\tilde{\mathbf{s}})$  with eigenvalue  
337  $\lambda_k(\tilde{\mathbf{s}})$  and  $\mathbf{v}_k^\dagger(\tilde{\mathbf{s}})$  is the adjoint eigenvector, chosen so that  
338  $\mathbf{v}_k(\tilde{\mathbf{s}}) \cdot \mathbf{v}_l^\dagger(\tilde{\mathbf{s}}) = \delta_{kl}$ . In terms of these eigenvectors and ei-  
339 genvalues, Eq. (8) becomes

$$\delta\mathbf{a}(t) = \sum_k \exp(\lambda_k(\tilde{\mathbf{s}})t) \mathbf{v}_k(\tilde{\mathbf{s}}) \mathbf{v}_k^\dagger(\tilde{\mathbf{s}}) \cdot \delta\mathbf{a}(0). \quad (9)$$

341 Since Eq. (2) admits an attractor,  $M$  of the eigen-  
342 values are zero—these correspond to perturbations  
343 along the attractor—and the rest are negative. For  
344 convenience, we rank the eigenvectors in order of  
345 decreasing eigenvalue, so  $\mathbf{v}_k(\tilde{\mathbf{s}})$  and  $\mathbf{v}_k^\dagger(\tilde{\mathbf{s}})$ ,  $k = 1, \dots, M$ ,  
346 are the eigenvectors and adjoint eigenvectors whose ei-  
347 genvalues are zero. (Interestingly, the first  $M$  eigenvec-  
348 tors,  $\mathbf{v}_k$ , can be expressed in terms of  $\mathbf{g}$ : combining Eqs.  
349 (3) and (7), it is not hard to show that  $\mathbf{v}_k(\mathbf{s}) = \partial_{s_k} \mathbf{g}(\mathbf{s})$ .) In  
350 the limit that  $t \rightarrow \infty$ , the only terms in Eq. (9) that  
351 survive are the ones with  $\lambda_k = 0$ ; we thus have

$$\lim_{t \rightarrow \infty} \delta\mathbf{a}(t) = \sum_{k=1}^M \mathbf{v}_k(\tilde{\mathbf{s}}) \mathbf{v}_k^\dagger(\tilde{\mathbf{s}}) \cdot \delta\mathbf{a}(0). \quad (10)$$

353 The value of  $\delta\mathbf{a}(\infty)$  given in Eq. (10) tells us the final  
354 point on the attractor. Knowing  $\delta\mathbf{a}(\infty)$  would allow us  
355 to find  $\hat{\mathbf{s}}$  in terms of  $\tilde{\mathbf{s}}$ . However, it is more convenient to  
356 choose  $\tilde{\mathbf{s}}$  so that  $\delta\mathbf{a}(\infty) = 0$ , because in that case,  $\hat{\mathbf{s}} = \tilde{\mathbf{s}}$ .  
357 The condition that  $\delta\mathbf{a}(\infty) = 0$  is that  $\mathbf{v}_k^\dagger(\tilde{\mathbf{s}}) \cdot \delta\mathbf{a}(0) = 0$   
358 for  $k = 1, \dots, M$ . Using Eqs. (4) and (5) to express  $\delta\mathbf{a}(0)$   
359 in terms of  $\mathbf{f}(\mathbf{s})$  and  $\mathbf{N}(\mathbf{s})$ , and replacing  $\tilde{\mathbf{s}}$  with  $\hat{\mathbf{s}}$ , this  
360 condition translates into  $M$  equations,

$$\mathbf{v}_k^\dagger(\hat{\mathbf{s}}) \cdot [\mathbf{f}(\mathbf{s}) + \mathbf{N}(\mathbf{s}) - \mathbf{g}(\mathbf{s})] = 0 \quad (11)$$

362 for  $k = 1, \dots, M$ .

363 To find  $\hat{\mathbf{s}}$  in terms of  $\mathbf{s}$ , we let  $\hat{\mathbf{s}} = \mathbf{s} + \delta\mathbf{s}$ ; term by term,  
364 this means that  $\hat{s}_k = s_k + \delta s_k$ . Expanding Eq. (11) to first  
365 order in  $\delta s$ , we arrive at the set of equations

$$\mathbf{v}_k^\dagger(\mathbf{s}) \cdot \mathbf{N}(\mathbf{s}) + \mathbf{v}_k^\dagger(\mathbf{s}) \cdot [\mathbf{f}(\mathbf{s}) - \mathbf{g}(\mathbf{s})] + \sum_l \delta s_l \partial_{s_l} \left( \mathbf{v}_k^\dagger(\hat{\mathbf{s}}) \cdot [\mathbf{f}(\mathbf{s}) + \mathbf{N}(\mathbf{s}) - \mathbf{g}(\hat{\mathbf{s}})] \right)_{\hat{\mathbf{s}}=\mathbf{s}} = 0. \quad (12)$$

367 If the term  $\mathbf{v}_k^\dagger(\mathbf{s}) \cdot [\mathbf{f}(\mathbf{s}) - \mathbf{g}(\mathbf{s})]$  does *not* vanish for all  $s$ ,  
368 then, for some  $s$ ,  $\delta s$  will be nonzero in the limit that the  
369 noise goes to zero, and the network will produce biased  
370 estimates. Conversely, if it does vanish, then the network  
371 estimator will be unbiased. We assume an unbiased

estimator (a condition that must be checked for indi- 372  
vidual networks), which requires that 373

$$\mathbf{v}_k^\dagger(\mathbf{s}) \cdot [\mathbf{f}(\mathbf{s}) - \mathbf{g}(\mathbf{s})] = 0 \quad (13)$$

for  $k = 1, \dots, M$ . If Eq. (13) is satisfied, then Eq. (12) 375  
implies that, for small  $\mathbf{N}$ ,  $\delta\mathbf{s} \sim \mathbf{N}$ . Thus, the term 376  
 $\delta s_l \partial_{s_l} \mathbf{v}_k^\dagger(\mathbf{s}) \cdot \mathbf{N}(\mathbf{s})$  that appears in Eq. (12) is  $\mathcal{O}(\mathbf{N}^2)$  and 377  
can be ignored. With this simplification, we find that  $\delta\mathbf{s}$  378  
is given by 379

$$\delta s_k = \sum_l [\mathbf{v}_k^\dagger(\mathbf{s}) \cdot \partial_{s_l} \mathbf{f}(\mathbf{s})]^{-1} \mathbf{v}_l^\dagger(\mathbf{s}) \cdot \mathbf{N}(\mathbf{s}). \quad (14)$$

In this expression, and in what follows, we are using a 381  
shorthand notation for the inverse of a matrix: 382  
 $[A_{kl}]^{-1} \equiv [A^{-1}]_{kl}$ . Thus,  $[\mathbf{v}_k^\dagger(\mathbf{s}) \cdot \partial_{s_l} \mathbf{f}(\mathbf{s})]^{-1}$  is the  $kl$ th 383  
component of the inverse of the matrix  $\mathbf{v}_k^\dagger(\mathbf{s}) \cdot \partial_{s_l} \mathbf{f}(\mathbf{s})$ . To 384  
derive Eq. (14) we used  $(\partial_{s_l} \mathbf{v}_k^\dagger) \cdot [\mathbf{f} - \mathbf{g}] - \mathbf{v}_k^\dagger \cdot \partial_{s_l} \mathbf{g} =$  385  
 $-\mathbf{v}_k^\dagger \cdot \partial_{s_l} \mathbf{f}$ , which follows from Eq. (13). 386

Using Eq. (14), it is straightforward to compute the 387  
covariance matrix that determines the error in the esti- 388  
mate of the angles, and we find that 389

$$\langle \delta s_k \delta s_l \rangle = \left[ \partial_{s_k} \mathbf{f}(\mathbf{s}) \cdot \left( \sum_{ij} \mathbf{v}_i^\dagger(\mathbf{s}) [\mathbf{v}_i^\dagger(\mathbf{s}) \cdot \mathbf{R}(\mathbf{s}) \cdot \mathbf{v}_j^\dagger(\mathbf{s})]^{-1} \mathbf{v}_j^\dagger(\mathbf{s}) \cdot \partial_{s_l} \mathbf{f}(\mathbf{s}) \right) \right]$$

where  $\mathbf{R}(\mathbf{s})$  is the noise covariance matrix, 391

$$\mathbf{R}(\mathbf{s}) \equiv \langle \mathbf{N}(\mathbf{s}) \mathbf{N}(\mathbf{s}) \rangle.$$

Because we now have two covariance matrices,  $\mathbf{R}$  and 393  
 $\langle \delta s \delta s \rangle$ , we will consistently refer to  $\mathbf{R}$  as the noise 394  
covariance matrix and  $\langle \delta s \delta s \rangle$  simply as the covariance 395  
matrix. 396

As discussed, above, our measure of the quality of the 397  
estimator is the determinant of the covariance matrix. 398  
To find the value of  $\mathbf{v}_k^\dagger$  that minimizes this determinant, 399  
we use the relation  $(d/dx) \log \det \mathbf{A} = \text{Tr}\{\mathbf{A}^{-1} \cdot d\mathbf{A}/dx\}$  400  
where  $\text{Tr}$  denotes trace. After straightforward, but ted- 401  
ious, algebra, we find that 402

$$\frac{d \log \det \langle \delta s \delta s \rangle}{d\mathbf{v}_k^\dagger} = 2 \sum_i \left[ \mathbf{R} \cdot \mathbf{v}_i^\dagger [\mathbf{v}_i^\dagger \cdot \mathbf{R} \cdot \mathbf{v}_k^\dagger]^{-1} - [\partial_{s_k} \mathbf{f} \cdot \mathbf{v}_i^\dagger]^{-1} \partial_{s_l} \mathbf{f} \right]. \quad (15)$$

The determinant of the covariance matrix is minimized 404  
when the right hand side of Eq. (15) is zero. This occurs 405  
when 406

$$\mathbf{v}_k^\dagger \propto \mathbf{R}^{-1} \cdot \partial_{s_k} \mathbf{f}, \quad (16)$$

at which point the covariance matrix simplifies to 408

$$\langle \delta s_k \delta s_l \rangle = [\partial_{s_k} \mathbf{f} \cdot \mathbf{R}^{-1} \cdot \partial_{s_l} \mathbf{f}]^{-1}.$$

Thus, whenever Eqs. (13) and (16) are satisfied, the 410  
nonlinear recurrent network given in Eq. (2) leads to a 411  
covariance matrix such that 412

$$\det\langle\delta\mathbf{s}\delta\mathbf{s}\rangle = \frac{1}{\det[\partial_s\mathbf{f}(\mathbf{s}) \cdot \mathbf{R}^{-1}(\mathbf{s}) \cdot \partial_s\mathbf{f}(\mathbf{s})]}.$$

414 The above analysis provided us with the best network  
415 estimator within the class of attractor networks,  
416 assuming the noise is small. How good is this network,  
417 and how small must the noise be? To answer these  
418 questions, we use the fact that the lower bound on the  
419 determinant of the covariance matrix is given by the  
420 inverse of the Fisher Information [5],

$$\det\langle\delta\mathbf{s}\delta\mathbf{s}\rangle \geq \frac{1}{\det I}, \quad (17)$$

422 where  $I$  is the Fisher information,

$$I_{kl} = \left\langle -\frac{\partial^2}{\partial s_k \partial s_l} \log P(\mathbf{a}(t=0)|\mathbf{s}) \right\rangle. \quad (18)$$

424 Eq. (17) is the multi-dimensional analog of the Cramér-  
425 Rao bound.

426 To compute the Fisher information, Eq. (18), we need  
427 to know the distribution of the noise; i.e., we need to  
428 know the explicit form of  $P(\mathbf{a}(0)|\mathbf{s})$ . Let us consider two  
429 types of noise: Gaussian with an arbitrary correlation  
430 matrix, for which

$$P(\mathbf{a}(0)|\mathbf{s}) = \frac{\exp[-(\mathbf{a}(0) - \mathbf{f}(\mathbf{s})) \cdot \mathbf{R}^{-1}(\mathbf{s}) \cdot (\mathbf{a}(0) - \mathbf{f}(\mathbf{s})) / 2]}{[(2\pi)^N \det \mathbf{R}(\mathbf{s})]^{1/2}} \quad (19)$$

432 and Poisson with uncorrelated noise, for which

$$P(\mathbf{a}(0)|\mathbf{s}) = \prod_i \frac{f_i(\mathbf{s})^{a_i(0)} e^{-f_i(\mathbf{s})}}{a_i(0)!}. \quad (20)$$

434 In Eq. (19),  $a_i$  is firing rate, while in Eq. (20),  $a_i$  is the  
435 number of spikes in an interval. For the Poisson distri-  
436 bution the mean value of  $\mathbf{a}(0)$  is  $\mathbf{f}$ , and the noise  
437 covariance matrix is given by

$$\langle (a_i(0) - f_i)(a_j(0) - f_j) \rangle_{\text{Poisson}} = f_i \delta_{ij} \equiv R_{ij},$$

439 where the subscript ‘‘Poisson’’ indicates an average over  
440 the probability distribution given in Eq. (20). Note that  
441 we are using the symbol  $\mathbf{R}$  for the noise covariance  
442 matrix of both the Gaussian and Poisson distributions;  
443 which distribution we mean should be clear from the  
444 context.

445 The Fisher information, Eq. (18), for the two cases is  
446 given by [1]

$$I_{kl, \text{Gaussian}} = \partial_{s_k} \mathbf{f} \cdot \mathbf{R}^{-1} \cdot \partial_{s_l} \mathbf{f} + \frac{1}{2} \text{Tr}\{\mathbf{R}^{-1} \cdot \partial_{s_k} \mathbf{R} \cdot \mathbf{R}^{-1} \cdot \partial_{s_l} \mathbf{R}\}, \quad (21a)$$

$$I_{kl, \text{Poisson}} = \partial_{s_k} \mathbf{f} \cdot \mathbf{R}^{-1} \cdot \partial_{s_l} \mathbf{f}. \quad (21b)$$

449 The trace term in Eq. (21a) is a nonnegative definite  
450 matrix with respect to the indices  $k$  and  $l$ , as is the first  
451 term on the right hand side of Eq. (21a). Thus,

$$\det[I_{\text{Gaussian}}] \geq \det[\partial_s \mathbf{f} \cdot \mathbf{R}^{-1} \cdot \partial_s \mathbf{f}],$$

$$\det[I_{\text{Poisson}}] = \det[\partial_s \mathbf{f} \cdot \mathbf{R}^{-1} \cdot \partial_s \mathbf{f}].$$

For noise in which the noise covariance matrix,  $\mathbf{R}$ ,  
depends on the stimulus,  $s$ , the network does not appear  
to reach the Cramér-Rao bound. However, for reason-  
able noise structures, it turns out that the second term in  
Eq. (21) vanishes as the noise goes to zero. Consider, for  
example, a covariance matrix in which the noise is  
modeled as an overall multiplicative term, which allows  
us to write  $\mathbf{R} = \epsilon \hat{\mathbf{R}}$  where  $\hat{\mathbf{R}}$  is independent of  $\epsilon$  and  $\epsilon$   
vanishes as the noise vanishes. With this change of  
variable, Eq. (21a) becomes

$$I_{kl, \text{Gaussian}} = \epsilon^{-1} \partial_{s_k} \mathbf{f} \cdot \hat{\mathbf{R}}^{-1} \cdot \partial_{s_l} \mathbf{f} + \frac{1}{2} \text{Tr}\{\hat{\mathbf{R}}^{-1} \cdot \partial_{s_k} \hat{\mathbf{R}} \cdot \hat{\mathbf{R}}^{-1} \cdot \partial_{s_l} \hat{\mathbf{R}}\}.$$

As the noise,  $\epsilon$ , goes to zero, the first term dominates  
and we recover the Cramér-Rao bound. For this to  
happen, we must have

$$\epsilon \ll \frac{\|\partial_{s_k} \mathbf{f} \cdot \hat{\mathbf{R}}^{-1} \cdot \partial_{s_l} \mathbf{f}\|}{\|\text{Tr}\{\hat{\mathbf{R}}^{-1} \cdot \partial_{s_k} \hat{\mathbf{R}} \cdot \hat{\mathbf{R}}^{-1} \cdot \partial_{s_l} \hat{\mathbf{R}}\}\|},$$

where  $\|\cdot\|$  denotes a norm over the indices  $k$  and  $l$  (the  
details of the norm are not important; we are interested  
only in scaling with the number of neurons). The term in  
the denominator, the trace term, typically scales as  $N$ ,  
the size of the noise covariance matrix. The scaling of  
the term in the numerator depends on the correlational  
structure. For uncorrelated noise, the numerator scales  
as  $\partial_{s_k} \mathbf{f} \cdot \partial_{s_l} \mathbf{f}$ , which is  $\mathcal{O}(N)$ . In this regime, the factor of  
 $N$  in the numerator and denominator cancel, and the  
network estimate is comparable to the Cramér-Rao  
bound whenever  $\epsilon \ll \mathcal{O}(1)$ . For correlated noise, how-  
ever, the numerator typically asymptotes to a constant  
for large  $N$  (see Appendix A). Thus, for noise that is  
correlated and depends on the stimulus, the network  
does not reach the Cramér-Rao bound unless  
 $\epsilon \ll \mathcal{O}(1/N)$ . Such noise is much smaller than is ob-  
served in practice, indicating that networks in the class  
considered here can be sub-optimal.

#### 4. Stimuli with variable reliability

The analysis in the previous section gave us an opti-  
mal network for fixed tuning curves and noise. In the  
real world, however, stimuli arrive with varying reli-  
ability: visual cues, for example, are more reliable in  
bright light than in dim light. Being able to deal with this  
situation is a difficult, yet critical, problem, because  
more than one cue may be available for inferring the  
value of perceptual variables. For instance, we often  
locate objects on the basis of their images and sounds,  
perceive the 3D structure of objects from binocular vi-

497 sion, extract shape from shading, determine structure  
498 from motion and perspective, and infer the position of  
499 our limbs from their image and proprioceptive feedback.  
500 Importantly, we perform these tasks accurately even  
501 though the reliability of any one of the cues can vary  
502 over a broad range.

503 Can a single network be optimal when the reliability  
504 of the cues is variable? The answer, of course, depends  
505 on how variability is encoded, but a reasonable  
506 assumption is that it is encoded in firing rate; that is, in  
507 the amplitude of the tuning curves,  $\mathbf{f}(\mathbf{s})$ . The question we  
508 address here is: if tuning curves are scaled by a constant  
509 factor to reflect the reliability of the cues, can the net-  
510 work still perform optimally?

511 Let us consider a network in which several stimuli are  
512 encoded in hills of activity, and the noise among dif-  
513 ferent hills is independent; networks of this type were  
514 shown by Deneve et al. [7] to be able to perform a broad  
515 range of computations optimally. In this type of net-  
516 work, the tuning curve,  $\mathbf{f}(\mathbf{s})$ , is concatenated into  $p$   
517 tuning curves,  $\mathbf{f}(\mathbf{s}) = (\mathbf{f}_1(\mathbf{s}), \mathbf{f}_2(\mathbf{s}), \dots, \mathbf{f}_p(\mathbf{s}))$ , one for each  
518 hill of activity (typically,  $p = M$ , but this is not neces-  
519 sarily the case). To mimic the variable reliability, we  
520 allow both the amplitudes of the individual tuning  
521 curves and the associated noise to be scaled. Given this  
522 scaling, the network is initialized via a slight modifica-  
523 tion of Eq. (4),

$$\mathbf{a}(0) = (\gamma_1 \mathbf{f}_1(\mathbf{s}) + \beta_1 \mathbf{N}_1(\mathbf{s}), \gamma_2 \mathbf{f}_2(\mathbf{s}) + \beta_2 \mathbf{N}_2(\mathbf{s}), \dots, \gamma_p \mathbf{f}_p(\mathbf{s}) + \beta_p \mathbf{N}_p(\mathbf{s})).$$

525 The independence of the noise among different hills  
526 implies that  $\langle \mathbf{N}_i \mathbf{N}_j \rangle = 0$  if  $i \neq j$ . Assuming that a net-  
527 work exists that is optimal when  $\gamma_i = \beta_i = 1$ , we would  
528 like to know whether the same network is also optimal  
529 when  $\gamma_i$  and  $\beta_i$  are not equal to 1, and if so, how  $\beta_i$   
530 should depend on  $\gamma_i$  to achieve optimality.

531 The two conditions for optimality are given in Eqs.  
532 (13) and (16). Consider first Eq. (13). In terms of the  
533 scaled, concatenated tuning curves, this equation be-  
534 comes

$$\mathbf{v}_k^\dagger(\mathbf{s}) \cdot [\gamma_i \mathbf{f}_i(\mathbf{s}) - \mathbf{g}_i(\mathbf{s})] = 0 \quad (22)$$

536 for  $i = 1, \dots, p$ . We will assume that Eq. (22) holds for  
537 all  $\gamma_i$ ; this would be the case, for instance, if  $\mathbf{v}_k^\dagger(\mathbf{s})$  were an  
538 odd function of its components and  $\mathbf{f}_i(\mathbf{s})$  and  $\mathbf{g}_i(\mathbf{s})$  were  
539 even functions. With this assumption, the network is  
540 optimal if Eq. (16) is satisfied. When  $\gamma_i = \beta_i = 1$ , Eq.  
541 (16) can be written

$$\langle \mathbf{N} \mathbf{N} \rangle \cdot \mathbf{v}_k^\dagger = c_k \partial_{s_k} \mathbf{f}, \quad (23)$$

543 where the  $c_k$  are a set of arbitrary constants and we used  
544  $\mathbf{R} = \langle \mathbf{N} \mathbf{N} \rangle$ . Since the noise associated with the different  
545 tuning curves are independent, Eq. (23) breaks up into  $p$   
546 equations, one for each set of tuning curves,

$$\langle \mathbf{N}_i \mathbf{N}_i \rangle \cdot \mathbf{v}_k^\dagger = c_k \partial_{s_k} \mathbf{f}_i. \quad (24)$$

Scaling  $\mathbf{f}_i$  by  $\gamma_i$  and  $\mathbf{N}_i$  by  $\beta_i$ , Eq. (24) becomes 548

$$\beta_i^2 \langle \mathbf{N}_i \mathbf{N}_i \rangle \cdot \mathbf{v}_k^\dagger = \gamma_i c_k \partial_{s_k} \mathbf{f}_i. \quad (25)$$

Eq. (25) is satisfied, and the network is optimal for 550  
tuning curves of arbitrary height, if  $\beta_i = \gamma_i^{1/2}$ . In other 551  
words, if the noise in the input to a network scales as the 552  
square root of the firing rate, then that network will be 553  
optimal, independent of the amplitude of the input. This 554  
is an important result, since the square root scaling is 555  
exactly what one finds for neurons that fire with Poisson 556  
statistics. Thus, Poisson statistics are in some sense 557  
optimal, at least for the kinds of networks we considered 558  
here, and nearly Poisson statistics, as are typically ob- 559  
served in cortical neurons [9,14,17], are nearly optimal. 560

This result confirms what we found in our previous 561  
study using computer simulations [7], which is that basis 562  
function networks exhibiting attractor dynamics can 563  
perform optimal estimation, and they can do so even 564  
when cues arrive with varying degrees of reliability. This 565  
result applies to any set of variables linked to one an- 566  
other through a nonlinear mapping and coded in the 567  
noisy activity of a population of neurons. What we 568  
showed here is that a network must exist that computes, 569  
from the noisy population codes, the optimal estimate of 570  
these variables, and does so regardless of their reliabil- 571  
ity—so long as the noise is Poisson and the reliability is 572  
encoded in firing rate. In such networks, the basis 573  
functions enforce the nonlinear mapping between the 574  
variables and the attractor dynamics ensures optimal 575  
statistical performance. 576

## 5. Improved efficiency network 577

For correlated, stimulus-dependent noise, the class of 578  
networks considered in the previous sections reach the 579  
Cramér-Rao bound only when the noise is extremely 580  
small, on the order of  $1/N$ . Are there networks that can 581  
do better? The analysis of Section 2 indicates that there 582  
are; all that is required is an optimal estimator,  $\hat{\mathbf{s}}(\mathbf{a})$ , and 583  
an attractor network whose time evolution preserves its 584  
inverse,  $\mathbf{a}(\hat{\mathbf{s}})$ . The surface  $\mathbf{a}(\hat{\mathbf{s}})$  may be highly curved, but 585  
in principle a network exists whose trajectories remain 586  
within the subspace  $\mathbf{a}(\hat{\mathbf{s}})$  if they start within that sub- 587  
space, as in Fig. 2. 588

To understand the properties of such a network, we 589  
consider the simple case of extracting the value of a 590  
stimulus that is encoded in the mean firing rate of a 591  
population of correlated neurons. For this case, we let 592  
the stimulus be one-dimensional—we refer to it simply 593  
as  $s$ —and we let  $a_i$  be the firing rate of the  $i$ th neuron. 594  
For the conditional distribution at time  $t = 0$ ,  $P(\mathbf{a}(0)|s)$ , 595  
we use the Gaussian distribution given in Eq. (19). (Note 596  
that the Gaussian distribution allows negative firing 597  
rates. While this is unrealistic, we use it because a more 598  
realistic probability distribution would greatly compli- 599

600 cate the analysis without changing the underlying re-  
601 sult.) In a slight departure from the previous section, we  
602 let the tuning curves be linear rather than hills of  
603 activity; that is,  $f_i(s) = \text{constant} \times s$ . For convenience,  
604 we set the constant to one, so  $f_i(s) = s$ . We let the noise  
605 covariance have the form

$$R_{ij} = \sigma^2(s)[\delta_{ij} + \rho(1 - \delta_{ij})]. \quad (26)$$

607 With this choice for the noise, the stimulus-dependent  
608 variance,  $\sigma^2$ , is the same for each neuron, and the  
609 pairwise correlation coefficient,  $\rho$ , is the same for each  
610 pair. For simplicity, we let  $\sigma^2$  be proportional to  $s$ :  
611  $\sigma^2(s) = \alpha s$ . This would be the case for Poisson-like  
612 neurons, in which the error in the estimate of firing rate  
613 increases with firing rate; for truly Poisson neurons,  $\alpha$   
614 would be one. The correlations in Eq. (26) could come  
615 from common input.

616 The Fisher information, Eq. (21a), is given by (see  
617 Appendix B)

$$I = \frac{1}{\alpha s} \frac{N}{N\rho + 1 - \rho} + \frac{N}{2s^2}, \quad (27)$$

619 where as usual, there are  $N$  neurons. The second term in  
620 Eq. (27) corresponds to the trace term in Eq. (21a). As  
621 discussed in the previous section, in the large  $N$  limit this  
622 term is negligible compared to the first term only if the  
623 noise,  $\alpha$ , is extremely small; for this example, it must be  
624 much smaller than  $2s/N\rho$ . Thus, unless  $\alpha \ll 2s/N\rho$ , the  
625 class of networks derived in the previous section will do  
626 poorly compared to the Cramér-Rao bound.

627 To see how to construct a more efficient network, we  
628 compute the maximum likelihood estimator. This is  
629 done by maximizing  $\log[P(\mathbf{a}(0)|s)]$ , the log likelihood of  
630 the conditional distribution. A straightforward calcula-  
631 tion (see Appendix B) yields

$$\frac{d \log P(\mathbf{a}(0)|s)}{ds} = \frac{N}{2s\sigma^2} \left[ \sigma^2 + \frac{s^2 - \bar{a}^2}{N\rho + 1 - \rho} - \frac{\overline{\delta a^2}}{1 - \rho} \right], \quad (28)$$

633 where  $\bar{a} \equiv N^{-1} \sum_i a_i(0)$  is the initial mean and  
634  $\overline{\delta a^2} \equiv N^{-1} \sum_i a_i^2(0) - \bar{a}^2$  is the initial variance.

635 Setting the right hand side of Eq. (28) to zero and  
636 solving for  $s$  in terms of  $\mathbf{a}$  yields the maximum likelihood  
637 estimator, which we denote  $\hat{s}_{\text{ML}}(\mathbf{a})$ . What does the sur-  
638 face  $\mathbf{a}(\hat{s}_{\text{ML}})$  look like; i.e., what is the shape of the sur-  
639 face in activity space that satisfies  $d \log P(\mathbf{a}|s)/ds = 0$ ?  
640 To answer this, it is convenient to make the orthogonal  
641 change of variables

$$\mathbf{a}(0) = \sum_{k=0}^{N-1} x_k \mathbf{u}_k, \quad (29)$$

643 where

$$\mathbf{u}_0 = N^{-1/2}(1, 1, \dots, 1), \quad (30)$$

i.e.,  $u_{0i} = N^{-1/2} \forall i$ , and the  $\mathbf{u}_k$  are orthogonal: 645  
 $\mathbf{u}_k \cdot \mathbf{u}_l = \delta_{kl}$ . With this change of variables, 646  
 $d \log P(\mathbf{a}(0)|s)/ds = 0$  when 647

$$\frac{x_0^2}{N\rho + 1 - \rho} + \sum_{k=1}^{N-1} \frac{x_k^2}{1 - \rho} = N\alpha s + \frac{N}{N\rho + 1 - \rho} s^2. \quad (31)$$

The surface associated with Eq. (31) is thus cigar 649  
shaped, with the long axis pointing in the  $\mathbf{u}_0$  direction 650  
and an aspect ratio of  $[(N\rho + 1 - \rho)/(1 - \rho)]^{1/2} \sim N^{1/2}$ . 651  
Thus, when  $N$  is large, the surface is extremely long and 652  
thin. This makes the curvature very tight, so it is not 653  
surprising that the linear approximation breaks down 654  
and the network derived using linear perturbation does 655  
not provide a good estimate unless  $\alpha \ll 1/N$ . 656

To understand how to derive a better network esti- 657  
mator, we need an expression for the maximum likeli- 658  
hood estimator. Setting the right hand side of Eq. (28) to 659  
zero and using the relation  $\sigma^2(s) = \alpha s$ , we see that, in the 660  
large  $N$  limit, this estimator is given by 661

$$\hat{s}_{\text{ML}}(\mathbf{a}) = \frac{\overline{\delta a^2}}{\alpha(1 - \rho)}. \quad (32)$$

As we show in Appendix B, in the large  $N$  limit,  $\hat{s}_{\text{ML}}(\mathbf{a})$  is 663  
unbiased and its variance is  $2s^2/N$ , the same as the 664  
Cramér-Rao bound. Thus, the estimator derived from 665  
maximum likelihood is efficient, in the sense that it 666  
reaches the Cramér-Rao bound. That  $\hat{s}_{\text{ML}}(\mathbf{a})$  is efficient 667  
is a peculiarity of high dimensional spaces: for corre- 668  
lated variables, in the large  $N$  limit, the mean has a 669  
variance that is  $\mathcal{O}(1)$  while the variance has a variance 670  
that is  $\mathcal{O}(1/N)$ . The maximum likelihood estimator gi- 671  
ven in Eq. (32) makes use of this fact, along with the fact 672  
that the variance scales with the mean. This result 673  
should dispel the myth that averaging large numbers of 674  
correlated neurons does not improve the estimate of 675  
correlated firing rates [14,21]—it does improve the esti- 676  
mate; one just has to compute the variance, not the 677  
mean. 678

Is there a neuronal network that can estimate  $s$  with a 679  
variance equal to the minimum,  $2s^2/N$ ? In principle, yes, 680  
but it requires nonlinear synapses. For instance, con- 681  
sider the set of network equations 682

$$\tau \frac{d\mathbf{a}}{dt} = \mathbf{u}_\perp - [\mathbf{a} - \mathbf{u}_0 \mathbf{u}_0 \cdot \mathbf{a}] \frac{\mathbf{u}_\perp \cdot \mathbf{a}}{\mathbf{a} \cdot \mathbf{a} - (\mathbf{u}_0 \cdot \mathbf{a})^2} - \mathbf{u}_0 (\mathbf{u}_0 \cdot \mathbf{a} - c_0),$$

where  $\mathbf{u}_0$  is given in Eq. (30) and  $\mathbf{u}_\perp$  is any vector 684  
orthogonal to  $\mathbf{u}_0$ , normalized so that  $\mathbf{u}_\perp \cdot \mathbf{u}_\perp = 1$ . It is 685  
not hard to show that this equation admits a line 686  
attractor. In particular, if  $\mathbf{a}(0) = \mathbf{f}(s) + \mathbf{N}(s)$ , then  $\mathbf{a}$  687  
asymptotes to the point  $N\overline{\delta a^2} \mathbf{u}_\perp + c_0 \mathbf{u}_0$  as  $t \rightarrow \infty$ . Once 688  
the network has asymptoted to that point,  $\overline{\delta a^2}$  is known, 689  
and thus so is the maximum likelihood estimate of  $s$ , 690  
 $\hat{s}_{\text{ML}}(\mathbf{a})$ , via see Eq. (32). 691

Unfortunately, it is not clear that such a network is 692  
biologically plausible. Nor is it clear that such a network 693

694 would generalize: we were able to find an analytic  
695 expression for  $\hat{s}_{ML}(\mathbf{a})$ , and thus a network that would  
696 compute it, only because we chose a very simple model.  
697 For more realistic, and thus more complex models, we  
698 do not know how easily an estimator can be found.  
699 Nevertheless, it may be possible to train an attractor  
700 network so that its trajectories are confined to the  
701 highly curved subspaces that arise when the covariance  
702 matrix depends on the stimulus.

## 703 6. Discussion

704 The brain has a hard job: it must store and manip-  
705 ulate vast quantities of information, it must do so  
706 quickly and accurately, and it must do so with under-  
707 lying elements—neurons—that are not very reliable. In  
708 other words, the brain must carry out complex compu-  
709 tations using populations of neurons that never fire  
710 precisely the same way more than once, even on iden-  
711 tical tasks. The question we asked in this paper was: how  
712 can biologically plausible networks carry out these tasks  
713 with as little information loss as possible?

714 To address this question, we focused on a particular  
715 class of networks: multi-dimensional attractor networks,  
716 which are recurrent networks that relax onto a line or  
717 higher dimensional manifold in activity space. We chose  
718 these networks for several reasons: they are biologically  
719 plausible, in the sense that they mimic the highly  
720 recurrent connectivity seen in cortex [2], there is exper-  
721 imental evidence for the existence of line-attractor net-  
722 works that code for head direction in rats [15,16], and  
723 they can perform a large range of computations [7].

724 We asked the following question: suppose a multi-  
725 dimensional attractor network is initialized with noisy  
726 input coding for a set of variables, and after initializa-  
727 tion it evolves noise-free. Can the network manipulate  
728 the encoded variables—carry out a computation—while  
729 extracting all the information contained in the noisy  
730 input? What we showed analytically is that the answer is  
731 yes, provided only that the noise in the input is small.  
732 The size of “small” turns out to depend on the structure  
733 of the noise. If the noise among the different input  
734 neurons is uncorrelated, or if it is correlated but inde-  
735 pendent of the encoded variables, then “small” is with  
736 respect to  $\mathcal{O}(1)$ . If the noise is correlated *and* depends on  
737 the encoded variables, then “small” is with respect to  
738  $\mathcal{O}(1/N)$  where  $N$  is the number of neurons. In the latter  
739 case, there may be a network that does compute opti-  
740 mally; indeed, the analysis in Sections 2 and 5 suggests  
741 that there is. However, we were not able to prove the  
742 existence of such an optimal network in general.

743 Constructing networks that can perform optimally  
744 for fixed input is valuable, but in the real world input  
745 often arrives with varying degrees of reliability. For  
746 example, if input codes for the position of an object on

747 the retina, that input will be reliable in bright light but  
748 unreliable in dim light. Perhaps surprisingly, it turns out  
749 that the networks we considered can perform optimally  
750 when cues that arrive with varying degrees of reliability,  
751 so long as two conditions are met: reliability is coded in  
752 the amplitude of the activity (e.g., the firing rate), with  
753 more reliable cues exhibiting large amplitudes, and the  
754 variance in the noise is proportional to the mean activ-  
755 ity. These are both characteristic of cortical neurons, for  
756 which the variance in spike count is approximately  
757 proportional to the mean [9,14,17]. Thus, cortical net-  
758 works may be able to make use of the natural variability  
759 in firing patterns to perform optimal computations.

760 There are two caveats to this study. The first is that  
761 multi-dimensional attractors are structurally unstable,  
762 in the sense that small perturbations in network  
763 parameters tend to cause systematic drift along the  
764 attractor [18,20]. If the drift is too fast, then the network  
765 can no longer act as an optimal estimator. However, if  
766 the drift is slow relative to the relevant timescale (often  
767 only a few hundred ms, but sometimes much longer),  
768 then it can be ignored. In addition, Wu and Amari [19]  
769 showed recently that the drift can be stabilized by suit-  
770 able synaptic facilitation.

771 The second caveat is that we considered noise-free  
772 evolution. In essence, we ignored internal sources of  
773 noise that are known to exist in biological networks,  
774 such as synaptic failures and stochastic ion channels.  
775 Thus, the deterministic network evolution, which we  
776 considered here, should be thought of as an approxi-  
777 mation to the true probabilistic evolution. If the internal  
778 noise is sufficiently small and/or well behaved, however,  
779 networks that are optimal for noise-free evolution  
780 should also be near-optimal for noisy evolution, as  
781 indicated in Fig. 3.

782 We have shown that biologically plausible recurrent  
783 networks can perform optimal computations with noisy

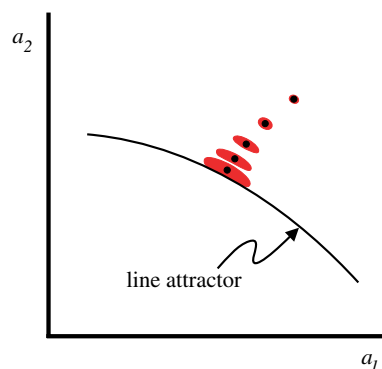


Fig. 3. Snapshots of deterministic (black) and probabilistic (red) trajectories in activity space. With no internal noise, the network evolves noise-free and follows the black points toward the line attractor. With internal noise, trajectories have a random component, as indicated by the expanding red blobs.

784 population codes, at least for uncorrelated or stimulus-  
785 independent noise in the input and for noise-free evo-  
786 lution. This is a first step toward understanding how  
787 spiking networks, which do not evolve noise-free, can  
788 perform optimal computations, and how they can do so  
789 when the noise is correlated and/or stimulus-dependent.

## 790 Acknowledgements

791 P.L. was supported by NIMH Grant #R01  
792 MH62447. A.P. and S.D. were supported by a fellow-  
793 ship from the Sloan Foundation, a young investigator  
794 award from ONR (N00014-00-1-0642) and a research  
795 grant from the McDonnell-Pew foundation.

## 796 Appendix A. Scaling of the Fisher information

797 The size of the noise for which the perturbatively  
798 derived network reaches the Cramér-Rao bound de-  
799 pends on how the first term in the Fisher information  
800 (Eq. (21a)) scales with  $N$ , the number of neurons. The  
801 second term in Eq. (21a) is, if nonzero, proportional to  
802  $N$ , so unless the first term also scales as  $N$ , the second  
803 will dominate. For correlated noise, the first term typi-  
804 cally asymptotes to a constant as  $N$  becomes large. We  
805 will not prove this, as there are counterexamples [1].  
806 Instead, we will motivate it using generic arguments,  
807 then illustrate those arguments with an example.

808 For simplicity, we consider the one-dimensional case,  
809 so the stimulus,  $s$ , is a scalar variable. Then, the first  
810 term in Eq. (21a), which we denote  $I_1$ , is given by

$$I_1 = \partial_s \mathbf{f} \cdot \mathbf{R}^{-1} \cdot \partial_s \mathbf{f}.$$

812 Our main tool for studying  $I_1$  is the eigenvalue expan-  
813 sion of  $\mathbf{R}^{-1}$ ,

$$\mathbf{R}^{-1} = \sum_k \frac{\mathbf{u}_k \mathbf{u}_k}{\lambda_k}, \quad (\text{A.1})$$

815 where the  $\mathbf{u}_k$  are the eigenvectors of  $\mathbf{R}$ , chosen to be  
816 orthogonal ( $\mathbf{u}_k \cdot \mathbf{u}_l = \delta_{kl}$ ), and the  $\lambda_k$  are the corre-  
817 sponding eigenvalues. Using Eq. (A.1),  $I_1$  becomes

$$I_1 = \sum_k \frac{(\partial_s \mathbf{f} \cdot \mathbf{u}_k)^2}{\lambda_k}. \quad (\text{A.2})$$

819 To make a crude estimate of scaling with  $N$ , we make  
820 the following observations: When there are correlations,  
821 many of the entries in  $R_{ij}$  are nonzero; consequently, the  
822  $\lambda_k$  scale as  $N$ . Because of the orthogonality conditions,  
823 the individual terms in  $\mathbf{u}_k$  scale as  $N^{-1/2}$ . Since there are  
824  $N$  terms in the dot-product,  $\partial_s \mathbf{f} \cdot \mathbf{u}_k$ , it scales as  $N^{1/2}$  and  
825 its square scales as  $N$ . The factors of  $N$  in the numerator  
826 and denominator thus cancel, and  $I_1$  scales as

$$I_1 = \sum_k \xi_k,$$

where  $\xi_k$  is  $\mathcal{O}(1)$ . Although there are  $N$  terms in the sum, 828  
only a finite number of them contribute. This is because 829  
 $\partial_s f_i$  is typically a smooth function of  $i$ , while the higher 830  
order eigenvectors are rapidly varying. Thus,  $I_1$  is  $\mathcal{O}(1)$ . 831

Let us see how this works for the particular example 832  
of a translation invariant noise covariance matrix, 833  
 $R_{ij} = R_{kl}$  if  $i - j = k - l$ . Typically,  $R_{ij}$  depends smoothly 834  
on the difference  $i - j$ , except when  $i - j = 0$  (as  $R_{ii}$  is the 835  
variance). We thus write 836

$$R_{jl} = r_0 \delta_{jl} + r_{j-l},$$

where  $r_j$  is a smooth function of  $j$ . The eigenvectors of 838  
 $R_{jl}$  are exponentials; letting  $u_{kj}$  be the  $j$ th component of 839  
 $\mathbf{u}_k$ , we have: 840

$$u_{kj} = \frac{e^{2\pi i j k / N}}{N^{1/2}}.$$

Consequently, the eigenvalues are given by 842

$$\lambda_k = r_0 + N r(k), \quad (\text{A.3})$$

where  $r(k)$  is the discrete Fourier transform of  $r_j$ , 844

$$r(k) = \frac{1}{N} \sum_j r_j e^{2\pi i j k / N}.$$

Define also  $\partial_s f(k)$  as the discrete Fourier transform of 846  
 $\partial_s f_i(s)$ , 847

$$\partial_s f(k) = \frac{1}{N^{1/2}} \partial_s \mathbf{f} \cdot \mathbf{u}_k = \frac{1}{N} \sum_j \partial_s f_j e^{2\pi i j k / N}. \quad (\text{A.4})$$

Both  $r(k)$  and  $\partial_s f(k)$  are  $\mathcal{O}(1)$ . 849

Combining Eqs. (A.3) and (A.4) with (A.2), we arrive 850  
at 851

$$I_1 = \sum_k \frac{|\partial_s f(k)|^2}{r_0/N + r(k)}. \quad (\text{A.5})$$

In the limit of large  $N$ , we can replace the sum in Eq. 853  
(A.5) by an integral, yielding 854

$$I_1 = \int dk \frac{|\partial_s f(k)|^2}{r_0/N + r(k)}$$

which is clearly independent of  $N$  in the limit  $N \rightarrow \infty$ . 856

Although we have not proved that  $I_1$  is independent 857  
of  $N$  as  $N \rightarrow \infty$  in general (as, in fact, it is not), we have 858  
shown one common correlational structure for which  $I_1$  859  
does asymptote to a constant. In addition, using the 860  
eigenvector expansion for the covariance matrix, we 861  
argued that this is a relatively robust feature. It requires 862  
only that the eigenvalues of  $\mathbf{R}$  scale as  $N$  and that the dot 863  
product,  $\partial_s \mathbf{f} \cdot \mathbf{u}_k$ , makes a nonnegligible contribution to 864  
 $I_1$  only for a finite set of  $k$ , even as  $N$  goes to  $\infty$ . 865

866 **Appendix B. Correlated neurons with firing rate propor-**  
867 **tional to the mean**

868 In this Appendix we fill in many of the missing steps  
869 in Section 5. We (1) compute the Fisher information for  
870 a Gaussian distribution with noise covariance matrix  
871 given in Eq. (26), (2) compute the derivative of the log  
872 likelihood, Eq. (28), and (3) show that the maximum  
873 likelihood estimator is unbiased and efficient (i.e., it  
874 reaches the Cramér-Rao bound).

875 All of these results rely on the properties of the noise  
876 covariance matrix. To streamline our calculations, we  
877 begin by expressing this matrix in terms of its eigen-  
878 vectors and eigenvalues. We start by rewriting slightly  
879 Eq. (A.1) to explicitly take into account the overall  
880 factor  $\sigma^2$ ,

$$\mathbf{R}^{-1} = \sigma^{-2}(s) \sum_k \frac{\mathbf{u}_k \mathbf{u}_k}{\lambda_k}, \quad (\text{B.1})$$

882 where  $\mathbf{R}$  is given in Eq. (26), the  $\mathbf{u}_k$  are the orthogonal  
883 eigenvectors of  $\mathbf{R}/\sigma^2$ , and the  $\lambda_k$  are the corresponding  
884 eigenvectors. (This is the same basis chosen in Eq. (29),  
885 so  $\mathbf{u}_0$  is given by Eq. (30).) It is not hard to show that  
886 there are two distinct eigenvalues,  $N\rho + 1 - \rho$ , which  
887 appears once and  $1 - \rho$ , which appears  $N - 1$  times. In a  
888 slight abuse of notation, we define

$$\lambda_0 \equiv N\rho + 1 - \rho, \quad (\text{B.2a})$$

$$\lambda_1 \equiv 1 - \rho. \quad (\text{B.2b})$$

891 Since  $f_i(s) = s^{\forall i}$ ,  $\mathbf{f}(s)$  can be expressed in terms of  $\mathbf{u}_0$  as  
$$\mathbf{f}(s) = sN^{1/2}\mathbf{u}_0. \quad (\text{B.3})$$

893 We can now compute the Fisher information, Eq.  
894 (21). Recalling that  $\sigma^2(s) = \alpha s$ , so that  $\partial_s \mathbf{R} = s^{-1} \mathbf{R}$ , and  
895 using Eq. (B.3) for  $\mathbf{f}(s)$ , we have

$$I = N\mathbf{u}_0 \cdot \mathbf{R}^{-1} \cdot \mathbf{u}_0 + \frac{1}{2s^2} \text{Tr}\{\mathbf{R}^{-1} \cdot \mathbf{R} \cdot \mathbf{R}^{-1} \cdot \mathbf{R}\}.$$

897 Using Eq. (B.1) for  $\mathbf{R}^{-1}$ , the orthogonality conditions on  
898 the  $\mathbf{v}_k$ , and the relation  $\text{Tr}\{\mathbf{I}\} = N$  where  $\mathbf{I}$  is the identity  
899 matrix (not to be confused with the Fisher information),  
900 it is trivial to show that the Fisher information reduces  
901 to the expression in Eq. (27).

902 To derive the maximum likelihood estimator, we  
903 differentiate  $\log \mathbf{P}(\mathbf{a}|s)$  with respect to  $s$ , where  $P(\mathbf{a}|s)$  is  
904 given in Eq. (19). (We use  $\mathbf{P}(\mathbf{a}|s)$  rather than  $\mathbf{P}(\mathbf{a}(0)|s)$   
905 for clarity.) Denoting differentiation with a prime and  
906 again applying the relation  $(d/dx)\log \det \mathbf{A} =$   
907  $\text{Tr}\{\mathbf{A}^{-1} d\mathbf{A}/dx\}$ , we have

$$\frac{d \log \mathbf{P}(\mathbf{a}|s)}{ds} = \mathbf{f}'(s) \cdot \mathbf{R}^{-1} \cdot (\mathbf{f}(s) - \mathbf{a}) + \frac{1}{2} (\mathbf{f}(s) - \mathbf{a}) \cdot \mathbf{R}^{-1'} \cdot (\mathbf{f}(s) - \mathbf{a}) + \frac{1}{2} \text{Tr}\{\mathbf{R}^{-1} \cdot \mathbf{R}'\}. \quad (\text{B.4})$$

909 Using  $\mathbf{R}^{-1'} = -s^{-1} \mathbf{R}^{-1}$ ,  $\mathbf{f}'(s) = s^{-1} \mathbf{f}(s)$  and, as above,  
910  $\mathbf{R}' = s^{-1} \mathbf{R}$  and  $\text{Tr}\{\mathbf{I}\} = N$ , Eq. (B.4) becomes

$$\frac{d \log \mathbf{P}(\mathbf{a}|s)}{ds} = \frac{1}{2s} [N + (\mathbf{f}(s) + \mathbf{a}) \cdot \mathbf{R}^{-1} \cdot (\mathbf{f}(s) - \mathbf{a})]. \quad (\text{B.5})$$

Using Eq. (B.1), it is straightforward to show that  $\mathbf{R}$  can  
be recast in the form

$$\mathbf{R}^{-1} = \sigma^{-2}(s) \left[ \mathbf{u}_0 \mathbf{u}_0 \left( \frac{1}{\lambda_0} - \frac{1}{\lambda_1} \right) + \frac{\mathbf{I}}{\lambda_1} \right]. \quad (\text{B.6})$$

Inserting Eq. (B.6) into (B.5) and performing a small  
amount of algebra, we arrive at

$$\frac{d \log \mathbf{P}(\mathbf{a}|s)}{ds} = \frac{N}{2s\sigma^2} \left[ \sigma^2 + \frac{\mathbf{f}(s) \cdot \mathbf{f}(s) - \mathbf{a} \cdot \mathbf{a}}{N\lambda_1} + \left( \frac{1}{\lambda_0} - \frac{1}{\lambda_1} \right) \frac{(\mathbf{f}(s) \cdot \mathbf{u}_0)^2 - (\mathbf{a} \cdot \mathbf{u}_0)^2}{N} \right]. \quad (\text{B.7})$$

To simplify this expression, as in the main text we define  
 $\bar{a} = N^{-1} \sum_i a_i$  and  $\overline{\delta a^2} = N^{-1} \sum_i a_i^2 - \bar{a}^2$ . Then, using  
these definitions and Eq. (30) for  $\mathbf{u}_0$ , we have  $\mathbf{f} \cdot \mathbf{f} = Ns^2$ ,  
 $\mathbf{f} \cdot \mathbf{u}_0 = N^{1/2}s$ ,  $\mathbf{a} \cdot \mathbf{u}_0 = N^{1/2}\bar{a}$ , and  $\mathbf{a} \cdot \mathbf{a} = N(\overline{\delta a^2} + \bar{a}^2)$ .  
With these relations, Eq. (B.7) becomes

$$\frac{d \log \mathbf{P}(\mathbf{a}|s)}{ds} = \frac{N}{2s\sigma^2} \left[ \sigma^2 + \frac{s^2 - \bar{a}^2}{\lambda_0} - \frac{\overline{\delta a^2}}{\lambda_1} \right]. \quad (\text{B.8})$$

When the definitions of  $\lambda_0$  and  $\lambda_1$  (Eq. (B.2)) are applied  
to Eq. (B.8), that equation becomes identical to the  
expression in Eq. (28).

Finally, we show that the maximum likelihood esti-  
mate, Eq. (32), is unbiased and efficient when  $N$  is large.  
We start with the mean,

$$\langle \hat{s}_{\text{ML}}(\mathbf{a}) \rangle = \frac{1}{\alpha(1-\rho)} \left[ \frac{1}{N} \sum_i \langle a_i^2 \rangle - \frac{1}{N^2} \sum_{ij} \langle a_i a_j \rangle \right],$$

where the angle brackets denote an average with respect  
to the Gaussian probability distribution given in Eq.  
(19). Because the distribution is Gaussian, the averages  
are trivial, and we have

$$\langle \hat{s}_{\text{ML}}(\mathbf{a}) \rangle = \frac{1}{\alpha(1-\rho)} \left[ \frac{1}{N} \sum_i R_{ii} - \frac{1}{N^2} \sum_{ij} R_{ij} \right].$$

Using Eq. (26), the first term inside the brackets is  $\sigma^2(s)$   
and the second term is  $\sigma^2(s)(1-\rho)(1-1/N)$ . Thus, in  
the large  $N$  limit, the terms inside the brackets reduce to  
 $\sigma^2(s)(1-\rho)$ , and

$$\langle \hat{s}_{\text{ML}}(\mathbf{a}) \rangle = \frac{\sigma^2(s)}{\alpha}.$$

Finally, using  $\sigma^2(s) = \alpha s$ , we see that  $\langle \hat{s}_{\text{ML}}(\mathbf{a}) \rangle = s$ , so the  
maximum likelihood estimator is unbiased.

The variance of  $\hat{s}_{\text{ML}}(\mathbf{a})$ , denoted  $\langle \delta \hat{s}_{\text{ML}}^2(\mathbf{a}) \rangle \equiv$   
 $\langle \hat{s}_{\text{ML}}(\mathbf{a})^2 \rangle - \langle \hat{s}_{\text{ML}}(\mathbf{a}) \rangle^2$ , can be computed in a similar  
manner,

$$\langle \delta s_{\text{ML}}^2(\mathbf{a}) \rangle = \frac{1}{\alpha^2(1-\rho)^2} \left[ \frac{1}{N^2} \sum_{ij} \left[ \langle a_i^2 a_j^2 \rangle - R_{ii} R_{jj} \right] - \frac{1}{N^3} \sum_{ijk} \langle a_i^2 a_j a_k \rangle + \frac{1}{N^4} \sum_{ijkl} \langle a_i a_j a_k a_l \rangle \right].$$

947 Using  $\langle a_i a_j a_k a_l \rangle = R_{ij} R_{kl} + R_{ik} R_{jl} + R_{il} R_{jk}$  for any  $i, j, k,$   
948 and  $l$ , this expression becomes

$$\langle \delta s_{\text{ML}}^2(\mathbf{a}) \rangle = \frac{2}{\alpha^2(1-\rho)^2} \left[ \frac{1}{N^2} \sum_{ij} R_{ij}^2 - \frac{2}{N^3} \sum_{ijk} R_{ij} R_{ik} + \frac{1}{N^4} \sum_{ijkl} R_{ij} R_{kl} \right].$$

950 Using Eq. (26), we find that, to lowest order in  $1/N$ , the  
951 terms inside the brackets add to  $\sigma^4(s)(1-\rho)^2/N$ . Con-  
952 sequently,

$$\langle \delta s_{\text{ML}}^2(\mathbf{a}) \rangle = \frac{2\sigma^4}{N\alpha^2}.$$

954 Finally, using  $\sigma^2 = \alpha s$ , we arrive at

$$\langle \delta s_{\text{ML}}^2(\mathbf{a}) \rangle = \frac{2s^2}{N},$$

956 which is the inverse of the Fisher information, Eq. (27),  
957 in the limit of large  $N$ .

## 958 References

- 959 [1] L.F. Abbott, P. Dayan, The effect of correlated variability on the  
960 accuracy of a population code, *Neural Comput.* 11 (1999) 91–101.  
961 [2] V. Braitenberg, A. Schüz, *Anatomy of the Cortex*, Springer-Verlag,  
962 Berlin, 1991.  
963 [3] N. Brunel, J.P. Nadal, Mutual information, Fisher information  
964 and population coding, *Neural Comput.* 10 (1998) 1731–1757.  
965 [4] M. Camperi, X.J. Wang, A model of visuospatial working  
966 memory in prefrontal cortex recurrent network and cellular  
967 bistability, *J. Comput. Neurosci.* 5 (1998) 383–405.  
968 [5] T.M. Cover, J.A. Thomas, *Elements of Information Theory*, John  
969 Wiley & Sons, New York, 1991.

- [6] S. Deneve, P.E. Latham, A. Pouget, Reading population codes: a  
970 neural implementation of ideal observers, *Nat. Neurosci.* 2 (1999)  
971 740–745. 972  
[7] S. Deneve, P.E. Latham, A. Pouget, Efficient computation and cue  
973 integration with noisy population codes, *Nat. Neurosci.* 4 (2001)  
974 826–831. 975  
[8] J. Droulez, A. Berthoz, A neural network model of sensoritopic  
976 maps with predictive short-term memory properties, *Proc. Natl.*  
977 *Acad. Sci.* 88 (1991) 9653–9657. 978  
[9] E.D. Gershon, M.C. Wiener, P.E. Latham, B.J. Richmond,  
979 Coding strategies in monkey VI and inferior temporal cortices,  
980 *J. Neurophysiol.* 79 (1998) 1135–1144. 981  
[10] J.J. Hopfield, Neural networks and physical systems with emer-  
982 gent collective computational abilities, *Proc. Natl. Acad. Sci.* 79  
983 (1982) 2554–2558. 984  
[11] J.J. Hopfield, Neurons with graded responses have collective  
985 computational properties like those of two-state neurons, *Proc.*  
986 *Natl. Acad. Sci.* 81 (1984) 3088–3092. 987  
[12] C.R. Laing, C.C. Chow, Stationary bumps in networks of spiking  
988 neurons, *Neural Comput.* 13 (2001) 1473–1494. 989  
[13] A. Pouget, K. Zhang, S. Deneve, P.E. Latham, Statistically  
990 efficient estimation using, population coding, *Neural Comput.* 10  
991 (1998) 373–401. 992  
[14] M.N. Shadlen, K.H. Britten, W.T. Newsome, J.A. Movshon, A  
993 computational analysis of the relationship between neuronal and  
994 behavioral responses to visual motion, *J. Neurosci.* 16 (1996)  
995 1486–1510. 996  
[15] J.S. Taube, R.U. Muller, J.B. Ranck Jr., Head-direction cells  
997 recorded from the post-subiculum in freely moving rats. I.  
998 Description and quantitative analysis, *J. Neurosci.* 10 (1990)  
999 420–435. 1000  
[16] J.S. Taube, R.U. Muller, J.B. Ranck Jr., Head-direction cells  
1001 recorded from the post-subiculum in freely moving rats. II. Effects  
1002 of environmental manipulations, *J. Neurosci.* 10 (1990) 436–447. 1003  
[17] D.J. Tolhurst, J.A. Movshon, A.F. Dean, The statistical reliability  
1004 of signals in single neurons in cat and monkey visual cortex, *Vis.*  
1005 *Res.* 23 (1983) 775–785. 1006  
[18] X.J. Wang, Synaptic reverberation underlying mnemonic persis-  
1007 tent activity, *Trends Neurosci.* 24 (2001) 455–463. 1008  
[19] S. Wu, S. Amari, *Neural Implementation of Bayesian Inference in*  
1009 *Population Codes*, Advances in Neural Information Processing  
1010 Systems, vol. 14, MIT Press, Cambridge, MA, 2002. 1011  
[20] K. Zhang, Representation of spatial orientation by the intrinsic  
1012 dynamics of the head-direction cell ensemble: a theory, *J.*  
1013 *Neurosci.* 16 (1996) 2112–2126. 1014  
[21] E. Zohary, M.N. Shadlen, W.T. Newsome, Correlated neuronal  
1015 discharge rate and its implications for psychophysical perfor-  
1016 mance, *Nature* 370 (1994) 140–143. 1017