Supplementary Materials

This section is organized into three parts. In the first, we show that when the likelihood function, $p(\mathbf{r}|s)$, belongs to the exponential family with linear sufficient statistics, optimal cue combination can be performed by a simple network in which firing rates from two population codes are combined linearly. Moreover, we show that the tuning curves of the two populations don't need to be identical, and that the responses both within and across populations don't need to be uncorrelated. In the second part, we consider the specific case of independent Poisson noise, which provides an example of a distribution belonging to the exponential family with linear sufficient statistics. We also consider a distribution that does not belong to the exponential family with linear sufficient statistics, namely, independent Gaussian noise with fixed variance. We show that, for this case, optimal cue combination requires a nonlinear combination of the population codes. In the third part, we describe in detail the parameters of the network of conductance-based integrate-and-fire neurons.

1. Probabilistic Population Codes for Optimal Cue Combination

1.1 Bayesian inference through linear combinations for the exponential family

Consider two population codes, $\mathbf{r}_1$ and $\mathbf{r}_2$ (both of which are vectors of firing rates), which code for the same stimulus, s. As described in the main text, this coding is probabilistic, so $\mathbf{r}_1$ and $\mathbf{r}_2$ are related to the stimulus via a likelihood function, $p(\mathbf{r}_1,\mathbf{r}_2|s)$. In a cue integration experiment, we need to construct a third population code, $\mathbf{r}_3$, related to $\mathbf{r}_1$ and $\mathbf{r}_2$ via some function: $\mathbf{r}_3=\mathbf{F}(\mathbf{r}_1,\mathbf{r}_2)$. Given this function, $p(\mathbf{r}_3|s)$ is given by

$$p\left(\mathbf{r}_3 \mid s\right)=\int p\left(\mathbf{r}_1,\mathbf{r}_2 \mid s\right)\delta\left(\mathbf{r}_3-F\left(\mathbf{r}_1,\mathbf{r}_2\right)\right)d\mathbf{r}_1 d\mathbf{r}_2. \qquad \text{(SM1)}$$

When $\mathbf{F}(\mathbf{r}_1,\mathbf{r}_2)$ is not invertible ($\mathbf{r}_3$ does not uniquely identify both $\mathbf{r}_1$ and $\mathbf{r}_2$), such a transformation could easily lose information. Our goal here is to find a transformation that does *not* lose information. Specifically, we want to choose $\mathbf{F}(\mathbf{r}_1,\mathbf{r}_2)$ so that

$$p\left(\mathbf{r}_3 \mid s\right)=p\left(\mathbf{F}\left(\mathbf{r}_1,\mathbf{r}_2\right) \mid s\right)\propto p\left(\mathbf{r}_1,\mathbf{r}_2 \mid s\right) \qquad \text{(SM2)}$$

where all terms are viewed as functions of s and the constant of proportionality is independent of s. If Equation (SM2) is satisfied, then Bayes' rule implies that $p(s|\mathbf{r}_3)$ is identical to $p(s|\mathbf{r}_1,\mathbf{r}_2)$, and one can use $\mathbf{r}_3$ rather than $\mathbf{r}_1$ and $\mathbf{r}_2$ without any loss of information about the stimulus. A function $\mathbf{F}(\mathbf{r}_1,\mathbf{r}_2)$ that satisfies Equation (SM2) is said to be *Bayes optimal*.

Clearly, the optimal function $\mathbf{F}(\mathbf{r}_1,\mathbf{r}_2)$ depends on the likelihood, $p(\mathbf{r}_1,\mathbf{r}_2|s)$. Here we show that if the likelihood lies in a particular family – exponential with linear sufficient statistics – then $\mathbf{F}(\mathbf{r}_1,\mathbf{r}_2)$ is linear in both $\mathbf{r}_1$ and $\mathbf{r}_2$. This makes optimal Bayesian inference particularly simple.

We start by considering the independent case, $p(\mathbf{r}_1,\mathbf{r}_2|s) = p(\mathbf{r}_1|s)p(\mathbf{r}_2|s)$; we generalize to the dependent case later on. As stated above, we consider likelihoods in the exponential family with linear sufficient statistics,

$$p\left(\mathbf{r}_k \mid s\right)=\frac{\phi_k\left(\mathbf{r}_k\right)}{\eta_k\left(s\right)}\exp\left(\mathbf{h}_k^{\mathbf{T}}\left(s\right)\mathbf{r}_k\right) \qquad \text{(SM3)}$$

where the superscript "T" denotes transpose and k=1, 2. Given this form for $p(\mathbf{r}_k|s)$, we show that if $\mathbf{h}_1(s)$ and $\mathbf{h}_2(s)$ can both be expressed as $\mathbf{h}_k(s)$=$\mathbf{A}_k\mathbf{b}(s)$ for some stimulus independent matrix $\mathbf{A}_k$ (i=1, 2), then optimal combination is performed by the linear function

$$\mathbf{r}_3=\mathbf{F}\left(\mathbf{r}_1,\mathbf{r}_2\right)=\mathbf{A}_1^{\mathbf{T}}\mathbf{r}_1+\mathbf{A}_2^{\mathbf{T}}\mathbf{r}_2 \qquad \text{(SM4)}$$

In other words, we show that when $\mathbf{r}_3$ is given by Equation (SM4) with $\mathbf{A}_1$ and $\mathbf{A}_2$ chosen correctly, Equation (SM2) is satisfied. Moreover, we show that the likelihood function $p(\mathbf{r}_3|s)$ lies in the same family of distributions as $p(\mathbf{r}_1|s)$ and $p(\mathbf{r}_2|s)$. This is important because it demonstrates that this approach – taking linear combinations of firing rates to perform optimal Bayesian inference – can be either repeated iteratively or cascaded from one population to the next. Finally, in section 1.2 below, we show that the stimulus dependent kernel functions, $\mathbf{h}_k(s)$, are related to the tuning curves of the populations, $\mathbf{f}_k(s)$, via the relationship

$$\mathbf{f}_k'\left(s\right) = \mathbf{\Sigma}_k\left(s\right)\mathbf{h}_k'\left(s\right) \tag{SM5}$$

where $\mathbf{\Sigma}_k\left(s\right)$ is the covariance matrix and $\mathbf{f}_k(s)$ is the tuning curve of the populations i=1,2.

To demonstrate these three properties, we use Equations (SM1) and (SM4), along with $\mathbf{h}_k(s)=\mathbf{A}_k\mathbf{b}(s)$, to compute the left hand side of Equation (SM2),

$$\begin{aligned} p\left(\mathbf{r}_3 \mid s\right) &= \int \frac{\phi_1\left(\mathbf{r}_1\right)\phi_2\left(\mathbf{r}_2\right)}{\eta_1\left(s\right)\eta_2\left(s\right)} \exp\left(\mathbf{b}^{\mathbf{T}}\left(s\right)\mathbf{A}_1^{\mathbf{T}}\mathbf{r}_1 + \mathbf{b}^{\mathbf{T}}\left(s\right)\mathbf{A}_2^{\mathbf{T}}\mathbf{r}_2\right)\delta\left(\mathbf{r}_3 - \mathbf{A}_1^{\mathbf{T}}\mathbf{r}_1 - \mathbf{A}_2^{\mathbf{T}}\mathbf{r}_2\right) d\mathbf{r}_1 d\mathbf{r}_2 \\ &= \int \frac{\phi_1\left(\mathbf{r}_1\right)\phi_2\left(\mathbf{r}_2\right)}{\eta_1\left(s\right)\eta_2\left(s\right)} \delta\left(\mathbf{r}_3 - \mathbf{A}_1^{\mathbf{T}}\mathbf{r}_1 - \mathbf{A}_2^{\mathbf{T}}\mathbf{r}_2\right) d\mathbf{r}_1 d\mathbf{r}_2 \exp\left(\mathbf{b}^{\mathbf{T}}\left(s\right)\mathbf{r}_3\right) \\ &= \frac{\phi_3\left(\mathbf{r}_3\right)}{\eta_3\left(s\right)} \exp\left(\mathbf{b}^{\mathbf{T}}\left(s\right)\mathbf{r}_3\right) \end{aligned} \tag{SM6}$$

where $\phi_3\left(\mathbf{r}_3\right) = \int \delta\left(\mathbf{r}_3 - \mathbf{A}_1^{\mathbf{T}}\mathbf{r}_1 - \mathbf{A}_2^{\mathbf{T}}\mathbf{r}_2\right)\phi_1\left(\mathbf{r}_1\right)\phi_2\left(\mathbf{r}_2\right) d\mathbf{r}_1 d\mathbf{r}_2$ and $\eta_3\left(s\right) = \eta_1\left(s\right)\eta_2\left(s\right)$. Meanwhile, the right hand side of Eq. (SM2) is given by

$$\begin{aligned} p\left(\mathbf{r}_1, \mathbf{r}_2 \mid s\right) &= \frac{\phi_1\left(\mathbf{r}_1\right)\phi_2\left(\mathbf{r}_2\right)}{\eta_1\left(s\right)\eta_2\left(s\right)} \exp\left(\mathbf{b}^{\mathbf{T}}\left(s\right)\mathbf{A}_1^{\mathbf{T}}\mathbf{r}_1 + \mathbf{b}^{\mathbf{T}}\left(s\right)\mathbf{A}_2^{\mathbf{T}}\mathbf{r}_2\right) \\ &= \frac{\phi_1\left(\mathbf{r}_1\right)\phi_2\left(\mathbf{r}_2\right)}{\eta_1\left(s\right)\eta_2\left(s\right)} \exp\left(\mathbf{b}^{\mathbf{T}}\left(s\right)\mathbf{r}_3\right). \end{aligned} \tag{SM7}$$

Comparing Equations (SM6) and (SM7), we see that both equations have the same dependence upon s, which implies that Equation (SM2) is satisfied, and thus information in preserved. Therefore, we conclude that optimal cue combination is performed by Equation (SM4), regardless of the choice of measure functions $\phi_1(\mathbf{r}_1)$ and $\phi_2(\mathbf{r}_2)$. While conditional independence of the two populations is assumed in the above derivation, this assumption is not necessary. Rather (as we show below), it is sufficient to assume that the joint distribution of $\mathbf{r}_1$ and $\mathbf{r}_2$ takes the form

$$p\left(\mathbf{r}_1,\mathbf{r}_2 \mid s\right)=\frac{\phi\left(\mathbf{r}_1,\mathbf{r}_2\right)}{\eta\left(s\right)}\exp\left(\mathbf{b}^{\mathbf{T}}\left(s\right)\mathbf{A}_1^{\mathbf{T}}\mathbf{r}_1+\mathbf{b}^{\mathbf{T}}\left(s\right)\mathbf{A}_2^{\mathbf{T}}\mathbf{r}_2\right) \tag{SM8}$$

for any $\phi\left(\mathbf{r}_1,\mathbf{r}_2\right)$ (see Equation (SM21)).

So far we have assumed that the likelihood, $p(\mathbf{r}|s)$, is a function *only* of the stimulus, s. In fact, the likelihood often depends on what is commonly called a *nuisance parameter* – something that affects the response distributions of the individual neural populations, but that the brain doesn't care about. For example, it is well known that contrast strongly affects the gain of the population and thus strongly affects the likelihood function. Since contrast represents the quality of the information about the stimulus but is otherwise independent of the actual value of the stimulus, the gain of the population, in this context, represents a nuisance parameter. To model this gain dependence, the likelihood functions for populations 1 and 2 should be written as $p(\mathbf{r}_1|s, g_1)$ and $p(\mathbf{r}_2|s, g_2)$ where g_k denotes gain of population k. Although we could apply our formalism and simply treat g_1 and g_2 as part of the stimulus, if we did that the likelihood for $\mathbf{r}_3$ would contain the term $\exp(\mathbf{b}^{\mathrm{T}}(\mathrm{s}, g_1, g_2)\ \mathbf{r}_3)$ (see Equation (SM8)). This is clearly inconvenient, because it means we would have to either know g_1 and g_2, or marginalize over these quantities, to extract the posterior distribution of the stimulus, s.

Fortunately, it is easy to show that this problem can be avoided if the nuisance parameter does not appear in the exponent, so that the likelihood is written

$$p\left(\mathbf{r} \mid s,g\right)=\phi\left(\mathbf{r},g\right)\exp\left(\mathbf{h}^{\mathbf{T}}\left(s\right)\mathbf{r}\right), \tag{SM9}$$

which is Equation (6) in the main text. When this is the case, either specifying g or multiplying by an arbitrary prior on g and marginalizing yields a conditional distribution $p(\mathbf{r}|s)$ which is in the desired family. If $\mathbf{h}(s)$ had been a function of g then this would not necessarily have been the case. Note that the normalization factor, $\eta(s,g)$, from Equation (SM8) is not present in Equation (SM9). This is because, marginalization of $p(\mathbf{r}|s,g)$ with respect to an arbitrary prior $p(g)$ will only leave the stimulus dependent kernel h(s) unchanged when the partition function, $\eta(s,g)$, factorizes into a term which depends only on s and a term which depends only on g, or equivalently, when

$$\begin{aligned} 0 &= \frac{d}{dg}\frac{d}{ds}\log\eta(s,g) \\ &= \frac{d}{dg}\frac{d}{ds}\log\int\phi(\mathbf{r},g)\exp\left(\mathbf{h}(s)^{\mathrm{T}}\mathbf{r}\right)d\mathbf{r}\,. \\ &= \frac{d}{dg}\mathbf{h}'(s)^{\mathrm{T}}\mathbf{f}(s,g) \end{aligned} \qquad \text{(SM10)}$$

However, when g is the gain, $\mathbf{f}(s,g)=g\bar{\mathbf{f}}(s)$, where $\bar{\mathbf{f}}(s)$ is independent of g. This implies

$$\begin{aligned} 0 &= \frac{d}{dg}\mathbf{h}'(s)^{\mathrm{T}}g\bar{\mathbf{f}}(s) \\ &= \mathbf{h}'(s)^{\mathrm{T}}\bar{\mathbf{f}}(s) \end{aligned}\,. \qquad \text{(SM11)}$$

However, since

$$\frac{d}{ds}\log\eta(s,g)=\mathbf{h}'(s)^{\mathrm{T}}g\bar{\mathbf{f}}(s) \qquad \text{(SM12)}$$

we can conclude (by combining SM11 and SM12) that when g is the gain, η(s,g) only factorizes when it is independent of s. Fortunately, this seemingly strict condition is satisfied in many biologically relevant scenarios. For example, this is the case if the function $\phi(\mathbf{r},g)$ and $\mathbf{h}(s)$ are both shift invariant, a standard assumption in theoretical studies of population codes. Here shift invariance means that

$$\begin{aligned} \phi(\mathbf{Sr},g) &= \phi(\mathbf{r},g) \\ \mathbf{h}(s+k\Delta s) &= \mathbf{Sh}(s) \end{aligned} \qquad \text{(SM13)}$$

where the *N* dimensional matrix **S** takes the form $s_{ij} = 1$ when mod(*i*-*j*-*k*,*N*)=0 and is zero otherwise, *k* is an integer which tells us how much the indices are shifted, and *N* is the number of neurons. Note that Equation (SM13) guarantees translation-invariant kernels,

$$h_i(s) = h(s - s_i), \tag{SM14}$$

and also translation invariant tuning curves and covariance matrices. Using the definition of the partition function, $\eta(s,g)$, and noting that det(**S**)=1, we see that

$$\begin{aligned}\eta(s,g) &= \int \phi(\mathbf{r},g)\exp(\mathbf{h}^{\mathrm{T}}(s)\mathbf{r})d\mathbf{r}\\ &= \int \phi(\mathbf{Sz},g)\exp\left(\mathbf{h}(s)^{\mathrm{T}}\mathbf{Sz}\right)d\mathbf{z}\\ &= \int \phi(\mathbf{z},g)\exp\left(\left(\mathbf{S}^{\mathrm{T}}\mathbf{h}(s)\right)^{\mathrm{T}}\mathbf{z}\right)d\mathbf{z}\\ &= \int \phi(\mathbf{z},g)\exp(\mathbf{h}^{\mathrm{T}}(s-k\Delta s)\mathbf{z})d\mathbf{z}\\ &= \eta(s-k\Delta s,g).\end{aligned} \tag{SM15}$$

Since *k* was arbitrary, $\eta(s,g)$ must be independent of *s*, and therefore is constant and any *g* dependence can be absorbed into $\phi(\mathbf{r},g)$.

Alternatively, we could also have concluded that $\eta(s,g)$ is independent of *s* by simply assuming that silence is uninformative, i.e. $p(s|\mathbf{r}=\mathbf{0},g)$ is equal to the prior $p(s)$, i.e.

$$\begin{aligned}p(s) &= p(s\,|\,\mathbf{r}=\mathbf{0},g)\\ &= \frac{\phi(\mathbf{0},g)\,p(s)}{\eta(s,g)}\left(\int \frac{\phi(\mathbf{0},g)\,p(s')}{\eta(s',g)}ds'\right)^{-1}.\\ &= \frac{p(s)}{\eta(s,g)}\left(\int \frac{p(s')}{\eta(s',g)}ds'\right)^{-1}\end{aligned} \tag{SM16}$$

Since the second term in the product on the right hand side is only a function of *g* equality holds only when $\eta(s,g)$ is independent of *s*. As shown in Fig. 3 in the main text, this condition can hold even when the tuning curves are not perfectly translation invariant.

1.2 Relationship between the tuning curves, the covariance matrix and the stimulus dependent kernel ***h****(s)*

In this section, we show that our approach works for a very wide range of tuning curves and covariance matrices. This follows from the combination of two facts 1) optimal combination via linear operations requires only that the stimulus dependent kernels, $\mathbf{h}_1(s)$ and $\mathbf{h}_2(s)$, be drawn from a common basis, i.e. $\mathbf{h}_k(s)=\mathbf{A}_k\mathbf{b}(s)$ and 2) the tuning curves and covariance matrix are related to the stimulus dependent kernels $\mathbf{h}(s)$ through a simple relationship (Equation (SM18) below). The first of these was shown in the previous section; the second we show here.

For any distribution of the form of Equation (SM9) , a relationship between the tuning curve and the stimulus dependent kernel can be obtained through a consideration of the derivative of the mean, $\mathbf{f}(s,g)$, with respect to the stimulus,

$$\begin{aligned}
\mathbf{f}'(s,g) &= \frac{d}{ds}\frac{\int \mathbf{r}\phi(\mathbf{r},g)\exp\left(\mathbf{h}^{\mathrm{T}}(s)\mathbf{r}\right)d\mathbf{r}}{\int \phi(\mathbf{r},g)\exp\left(\mathbf{h}^{\mathrm{T}}(s)\mathbf{r}\right)d\mathbf{r}} \\
&= \frac{\int \mathbf{r}\mathbf{r}^{\mathrm{T}}\mathbf{h}'(s)\phi(\mathbf{r},g)\exp\left(\mathbf{h}^{\mathrm{T}}(s)\mathbf{r}\right)d\mathbf{r}}{\int \phi(\mathbf{r},g)\exp\left(\mathbf{h}^{\mathrm{T}}(s)\mathbf{r}\right)d\mathbf{r}} \\
&\quad -\frac{\int \mathbf{r}\phi(\mathbf{r},g)\exp\left(\mathbf{h}^{\mathrm{T}}(s)\mathbf{r}\right)d\mathbf{r}}{\int \phi(\mathbf{r},g)\exp\left(\mathbf{h}^{\mathrm{T}}(s)\mathbf{r}\right)d\mathbf{r}}\frac{\int \mathbf{r}^{\mathrm{T}}\mathbf{h}'(s)\phi(\mathbf{r},g)\exp\left(\mathbf{h}^{\mathrm{T}}(s)\mathbf{r}\right)d\mathbf{r}}{\int \phi(\mathbf{r},g)\exp\left(\mathbf{h}^{\mathrm{T}}(s)\mathbf{r}\right)d\mathbf{r}} \qquad \text{(SM17)} \\
&= \left\langle \mathbf{r}\mathbf{r}^{\mathrm{T}}\right\rangle_{s,g}\mathbf{h}'(s)-\mathbf{f}(s,g)\mathbf{f}^{\mathrm{T}}(s,g)\mathbf{h}'(s) \\
&= \mathbf{\Sigma}(s,g)\mathbf{h}'(s).
\end{aligned}$$

Here $\mathbf{\Sigma}(s,g)$ is the covariance matrix and we have expressed the partition function $\eta(s,g)$ in its integral form. Clearly, since the covariance matrix may depend upon the stimulus, there is a great variety of tuning curves which may be optimally combined.

When the gain is present as a nuisance parameter, this relationship may also be used to demonstrate that the covariance matrix must be proportional to the gain. This is because we can rewrite Equation (SM18) as

$$\mathbf{h}'(s) = \boldsymbol{\Sigma}^{-1}(s,g)\mathbf{f}'(s,g) \quad \text{(SM18)}$$

This corresponds to Equation (7) in the main text. As noted above, the kernel $\mathbf{h}(s)$ must be independent of gain for the optimality of linear combinations. Since $\mathbf{f}'(s,g) = g\overline{\mathbf{f}}'(s)$ where $\overline{\mathbf{f}}(s)$ is independent of gain, this occurs if the covariance matrix is also proportional to the gain. Since the diagonal elements of the covariance matrix correspond to the variance, the constant of proportionality gives the Fano factor. The precise value of the constant of proportionality, and thus of the Fano factor, is not important, so long as it is independent of the gain.

1.3 Constraint on the posterior distribution over s

The basis from which $\mathbf{h}(s)$ is drawn not only determines whether or not two populations may be optimally combined, but also places some restrictions on the set of posterior distributions that can be represented. These restrictions, however, are quite weak in the sense that, for proper choices of the kernel **h**(s), a very wide range of posterior distributions can be represented.

For instance, consider the case in which the partition function, $\eta(s,g)$, is independent of *s*, so that the posterior distribution is simply

$$p(s\,|\,\mathbf{r}) \propto \exp\left(\mathbf{h}^{\mathrm{T}}(s)\mathbf{r}\right) \quad \text{(SM19)}$$

Thus, the log of the posterior is a linear combination of the functions that make up the vector $\mathbf{h}(s)$, and we may conclude that almost any posterior may be well approximated when this set of functions is "sufficiently rich." Of course, it is also possible to restrict the set of posterior distributions by an appropriate choice for $\mathbf{h}(s)$. For instance, if it is desirable that the posterior distribution be constrained to be Gaussian, we could simply restrict the basis of $\mathbf{h}(s)$ to the set quadratic functions of *s*. Equation (SM20) also indicates why gain is a particularly important nuisance parameter for distributions in this family: an increase in the amplitude of the population pattern of activity, **r**, leads to a significant increase in the sharpness of the posterior through the exponentiation.

1.4 Neural variability and the exponential family with linear sufficient statistics

In the above derivation we made no explicit assumptions regarding the covariance structure of the joint distribution of $\mathbf{r}_1$ and $\mathbf{r}_2$. Fortunately, as with the Gaussian distribution, there are members of this family of distributions which are capable of modeling the first-order and second-order statistics of any response distribution, as long at the tuning curves depend on the stimulus. A complete set of restrictions can be obtained through a consideration of the higher *s* derivatives of either the tuning curve or the partition function. However, as with the Gaussian distribution, these restrictions concern only the third and higher moments.

Together with Equation (SM18), these arguments indicate that a broad class of correlation structures between populations can also be incorporated into this encoding scheme. Specifically, in Equation (SM18) we did not specify whether or not the responses referred to one or two populations. Thus, the vector mean and covariance matrix of Equation (SM18), could have referred to a pair of correlated populations, i.e.,

$$\mathbf{f}(s,g)=\begin{bmatrix}\mathbf{f}_1(s,g)\\ \mathbf{f}_2(s,g)\end{bmatrix},\quad \boldsymbol{\Sigma}(s,g)=\begin{bmatrix}\boldsymbol{\Sigma}_{11}(s,g) & \boldsymbol{\Sigma}_{12}(s,g)\\ \boldsymbol{\Sigma}_{21}(s,g) & \boldsymbol{\Sigma}_{22}(s,g)\end{bmatrix},\quad \text{and}\quad \mathbf{h}(s)=\begin{bmatrix}\mathbf{h}_1(s)\\ \mathbf{h}_2(s)\end{bmatrix}. \text{(SM20)}$$

When this is the case, the two populations may be optimally combined, provided $\mathbf{h}_1(s)$ and $\mathbf{h}_2$(s), as obtained from Equations (SM18) and (SM21), are independent of *g* and linearly related, or more generally, drawn from a common basis.

2. An example showing explicitly that a linear combination is optimal (Poisson neurons), and a second example showing that a linear combination is not optimal (Gaussian neurons).

2.1 Independent Poisson neurons

We now consider an example of a distribution that belongs to the exponential family with linear sufficient statistics, namely the independent Poisson distribution. We also assume that the neurons have Gaussian tuning curves which are dense and translation invariant, i.e., $\sum_i f_i(s) = c$, where c is some constant. For this case, we have

$$
\begin{aligned}
p(\mathbf{r} \mid s, g) &= \prod_i \frac{(gf_i(s))^{r_i}}{r_i!} \exp(-gf_i(s)) \\
&= \exp\left(-g\sum_i f_i(s)\right) \prod_i \frac{(g)^{r_i}}{r_i!} \exp\left(r_i \log\left(f_i(s)\right)\right) \\
&= \exp(-gc)\left(\prod_i \frac{(g)^{r_i}}{r_i!}\right) \exp\left(\sum_i r_i \log\left(f_i(s)\right)\right) \\
&= \phi(\mathbf{r}, g) \exp\left(\mathbf{h}^{\mathrm{T}}(s)\mathbf{r}\right).
\end{aligned}
\qquad \text{(SM21)}
$$

Here $h_i(s) = \log(f_i(s))$ and g represents, as usual, the gain. Clearly, this likelihood function satisfies Equation (SM9). The stimulus dependent kernel $\mathbf{h}(s)$ in this case is simply the log of the tuning curves. Moreover, it is easy to show that if we marginalize out the gain we obtain a likelihood function, $p(\mathbf{r}|s)$, that satisfies Equation (SM3) regardless of the prior on g. In other words, for independent Poisson noise, optimal cue combination only involves linear combination of population pattern of activity. Moreover, for Gaussian tuning curves, the log of each $f_i(s)$ is quadratic in s, implying that the resulting posterior distribution is also a Gaussian with a variance, $\sigma^2(\mathbf{r})$, that is inversely proportional to the amplitude, i.e.,

$$
\frac{1}{\sigma^2(\mathbf{r})} = \sum_i \frac{r_i}{\sigma_i^2}. \qquad \text{(SM22)}
$$

Here, σ_i is the width of the i^{th} tuning curve.

2.2 Gaussian distributed neurons

In the above example, the assumption that the tuning curves are dense insures the parameter g can be marginalized without affecting the stimulus dependence of the likelihood function. This is not, however, always the case. For example, consider a population pattern of activity that has some stimulus-dependent mean $g\mathbf{f}(s)$ that is corrupted by independent Gaussian noise with a fixed variance σ^2, i.e.,

$$\begin{aligned} p(\mathbf{r} \mid s,g) &= \left|2\pi\sigma^2\right|^{-N/2} \exp\left(-\frac{1}{2\sigma^2}\left(\mathbf{r}-g\mathbf{f}(s)\right)^{\mathbf{T}}\left(\mathbf{r}-g\mathbf{f}(s)\right)\right) \\ &= \left|2\pi\sigma^2\right|^{-N/2} \exp\left(-\frac{\mathbf{r}^{\mathbf{T}}\mathbf{r}}{2\sigma^2}\right)\exp\left(-\frac{g^2\mathbf{f}^{\mathbf{T}}(s)\mathbf{f}(s)}{2\sigma^2}\right)\exp\left(\frac{g\mathbf{f}^{\mathbf{T}}(s)\mathbf{r}}{2\sigma^2}\right) \\ &= \left|2\pi\sigma^2\right|^{-N/2} \exp\left(-\frac{\mathbf{r}^{\mathbf{T}}\mathbf{r}}{2\sigma^2}\right)\exp\left(-\frac{g^2c}{2\sigma^2}\right)\exp\left(\frac{g\mathbf{f}^{\mathbf{T}}(s)\mathbf{r}}{2\sigma^2}\right) \\ &= \phi(\mathbf{r},g)\exp\left(g\mathbf{h}^{\mathbf{T}}(s)\mathbf{r}\right). \end{aligned} \qquad \text{(SM23)}$$

Here, $\mathbf{h}(s) = \mathbf{f}(s)/\sigma^2$ and the density of the tuning curves implies that $\mathbf{f}(s)^{\mathrm{T}}\mathbf{f}(s)$ is constant, independent of s. Unlike the independent Poisson case, it is now impossible to marginalize an arbitrary prior on the gain without affecting the stimulus dependence of the likelihood function. Of course, if the gains of two such populations are known, optimal Bayesian inference is performed by the linear operation,

$$\mathbf{r}_3 = g_1\mathbf{r}_1 + g_2\mathbf{r}_2. \qquad \text{(SM24)}$$

However, if the gains of both populations are not constant across trials, then the use of Equation (SM25) requires that the weights of the linear operation be changed on a trial by trial basis. That is, the gain of each population must be approximated, presumably from the activities of the populations themselves, such that

$$\mathbf{r}_3 = g_1(\mathbf{r}_1)\mathbf{r}_1 + g_2(\mathbf{r}_2)\mathbf{r}_2. \qquad \text{(SM25)}$$

Thus, for additive Gaussian noise, optimal cue combination cannot be performed by a linear operation.

3. Simulations with simplified neurons

This simulation (summarized in Fig.3 of the main text) illustrates the optimality of our approach for a network with different types of tuning curves in the input layers. Here we provide the details of those simulations.

The input contains three layers, with N neurons in each. One of the layers has Gaussian tuning curves; the other two have sigmoidal tuning curves; one monotonically increasing and the other monotonically decreasing. In all cases the noise is independent and Poisson. We generated the tuning curves using a two step process. First we generated the kernels, $\mathbf{h}_k$(s), for each input layer (k=1, 2 or 3) by combining linearly a set of basis functions, denoted $\mathbf{b}(s)$, using three distinct matrices, $\mathbf{A}_1$, $\mathbf{A}_2$ and $\mathbf{A}_3$. We then used the exponential of these kernels as input tuning curves. This is the correct choice of tuning curves when the noise is independent and Poisson.

The activity in the output layer was obtained by summing the input activity multiplied by the transpose of $\mathbf{A}_1$, $\mathbf{A}_2$ and $\mathbf{A}_3$ (as specified by Equations (SM4) and (SM29); see below). This procedure ensures that the kernel of the output layer is simply the basis set, $\mathbf{b}(s)$, used to generate the input kernels.

Generating the input kernel $\mathbf{h}_k(s)$ *and input tuning curves* $\mathbf{f}_k(s)$

We first generated a set of N basis functions defined as

$$b_i(s) = \log\left[M\left(\exp\left(-\frac{(s-s_i)^2}{2\sigma_i^2}\right)+c_i\right)\right]$$

with N = 51, M = 1, σ_i^2 = 32, c_i = 0.1 and s_i = -400+16*i. These basis functions were combined linearly to obtained the kernels in each of the input layers

$$\mathbf{h}_k(s) = \mathbf{A}_k\mathbf{b}(s) \tag{SM26}$$

where again k=1,2 and 3 (corresponding to the three input layers), and $\mathbf{A}_k$ is a matrix of coefficients specific to each input layer. The matrices $\mathbf{A}_k$ were obtained using linear regression with a regularizer (to keep the weights smooth and small). Specifically, we used

$$\mathbf{A}_k^T = \left[\mathbf{C}_\mathbf{b} + d\mathbf{I}\right]^{-1} \mathbf{C}_{\mathbf{bh}^*}^k \qquad \text{(SM27)}$$

where $\mathbf{C}_\mathbf{b}$ is the covariance matrix of the basis set **b** (across all values of *s*, assuming a uniform distribution over the range [-400, 400]), $\mathbf{C}^k_{\mathbf{bh}*}$ is the covariance between **b** and the target kernel **h*** for input layer *k*, and **I** is the identity matrix. The parameter *d* (the regularizer parameter) was set to 1.

The i^{th} target kernel in the Gaussian input layer was given by

$$h_i^*(s) = \log\left[M\left(\exp\left(-\frac{(s-s_i)^2}{2\sigma_i^2}\right) + d_i\right)\right]$$

with $N = 51$, $M = 1\pm0.5$, $\sigma_1^2 = 32\pm16$, $d_i = 0.1\pm0.1$ and $s_i = -400+16*i\pm4$. In all cases, the notation ± means that the parameters were drawn uniformly from the corresponding range of values (e.g. 32±16 = [16, 48]). The random components were added to introduce variability in the width, position, baseline and amplitude of the input tuning curves.

For the monotonic increasing sigmoidal input layer, the i^{th} target kernel was given by,

$$h_i^*(s) = \log\left[M\left(\frac{1}{1+\exp\left(-(s-s_i)/t\right)} + d_i\right)\right]$$

with $N = 51$, $M = 1\pm0.5$, $t = 32\pm16$, $d_i = 0.1\pm0.1$ and $s_i = -400+16*i\pm-4$. The same equation and parameters was used in the monotonic decreasing sigmoidal input layer, with a reversed sign in the exponential. The input tuning curves, $\mathbf{f}_k(s)$, were then obtained by taking the log of the input kernels, $\mathbf{h}_k(s)$.

Note that because of the approximation introduced by the linear regression step (Equations (SM27) and (SM28)), the input tuning curves are not exactly equal to the log of the target kernels. Nonetheless, they are quite close and, as a result, the tuning curves in the first input layer were indeed Gaussian, while the tuning curves is the other two layers were sigmoidal (see Fig. 3a in the main text).

Generating one trial

The activity in the input layers on each trial (see Fig. 3b in the main text) were obtained by drawing spike counts from a multivariate independent Poisson distribution with means $\mathbf{f}_k(s)$. The resulting activities, $\mathbf{r}_1$, $\mathbf{r}_2$ and $\mathbf{r}_3$, were then combined to obtain the activity in the output layer according to (see Equation (SM4)):

$$\mathbf{r}_o = \left[\mathbf{A}_1^{\mathbf{T}}\mathbf{r}_1 + \mathbf{A}_2^{\mathbf{T}}\mathbf{r}_2 + \mathbf{A}_3^{\mathbf{T}}\mathbf{r}_3\right]^+ \tag{SM28}$$

where the rectification $[]^+$ is defined as $[x]^+= \max(0,x)$. This rectification is used to ensure that all component of $\mathbf{r}_o$ are non-negative (as is the case for neural activity). This introduces a slight approximation in our scheme but, as can be seen from Fig. 3 in the main text, this has virtually no impact on our results.

Decoding the probability distributions

For a given pattern of activity, $\mathbf{r}_k$, in a layer k, the corresponding probability distributions is obtained through

$$p(s\,|\,\mathbf{r}_k) = \frac{1}{Z}\exp\left(\mathbf{h}_k^{\mathrm{T}}(s)\mathbf{r}_k\right)$$

Z is chosen to ensure that the integral of $p(s|\mathbf{r}_{\mathrm{k}})$ with respect to s is equal to 1. Note that in the output layer, the i^{th} kernel in the output layer is given by $b_i(s)$.

4. Simulations with conductance-based integrate-and-fire neurons

The objective of the simulations was to demonstrate that networks of conductance-based integrate-and-fire neurons can perform near-optimal Bayesian inference. As a case study, we used a cue combination task in which the network combines two cues whose reliabilities are systematically varied from trial to trial.

Network architecture

The network consists of two unconnected input layers and one output layer. Each input layer contains 252 independent excitatory neurons firing with near-Poisson statistics. The output layer contains 1008 excitatory and 252 inhibitory neurons. The preferred stimuli of the neurons in each layer are equally spaced and uniformly distributed. An excitatory neuron in the output layer receives 24 connections from neurons in each input layer, an inhibitory one receives 16. Connections are drawn randomly without replacement from a Gaussian probability distribution over the stimulus, centered at the preferred stimulus of the output neuron and with a width of σ_{kernel}. Specifically, the probability of making a connection from neuron j to neuron i, denoted p_{ij}, is given by

$$p_{ij} = \frac{\exp\left(-\frac{\left(s_i - s_j\right)^2}{2\sigma_{\text{kernel}}^2}\right)}{\sum_j \exp\left(-\frac{\left(s_i - s_j\right)^2}{2\sigma_{\text{kernel}}^2}\right)} \tag{SM29}$$

All connection strengths are equal and constant with value w for a given input layer.

Within the output layer there are two types of lateral connections: inhibitory to excitatory and excitatory to inhibitory. Each excitatory neuron receives 30 connections from inhibitory neurons, and each inhibitory neuron receives 40 connections from excitatory neurons. These are randomly drawn without replacement from a uniform distribution, and the connection strengths are all 1.

Input layers

Neurons in the input layers fire at a constant rate, except that there is a refractory period of 3 ms. More specifically, the probability of firing in any small interval *dt* is a constant times *dt*, except within 3 ms of a previous spike, in which case the probability of firing is 0. As a result, the variance of the spike counts of an input neuron across trials is approximately equal to their mean. The rates are obtained from a Gaussian distribution centered at a given stimulus, with width σ_{input}, plus a baseline set to a fraction of the amplitude (peak rate minus baseline rate),

$$\langle r_i \rangle_s = g\left(\exp\left(-\frac{\left(s-s_i\right)^2}{2\sigma_{\text{input}}^2}\right)+c\right). \tag{SM30}$$

We used σ_{input} and $c = 0.1$. The gain, *g*, is fixed on any given trial. In the case of the visual system, the amplitude would be related to the contrast of a presented image. The higher the contrast, the higher the input gain, the higher the output gain, and the less variable the estimate of stimulus.

Output layer

The output layer consists of conductance-based integrate-and-fire neurons. The membrane potential, $V_i\left(t\right)$, of output neuron *i* as a function of time *t* is described by

$$C\frac{dV_i}{dt} = -g_L\left(V_i - E_L\right) - g_{iE}\left(t\right)\left(V_i - E_E\right) - g_{iI}\left(t\right)\left(V_i - E_I\right) - g_{iA}\left(t\right)\left(V_i - E_A\right) \tag{SM31}$$

where *C* is the capacitance of the membrane and E_E, E_I, E_A, and E_L are reversal potentials. The conductance $g_{iE}\left(t\right)$ contains the contributions of the spikes from all excitatory presynaptic neurons. If neuron *i* is of type *a* (which can be *E* or *I*), then this conductance is given by

$$g_{ia}\left(t\right) = \sum_{jk} w_{ij}\bar{g}_{aj}\alpha_{\tau_a}\left(t - t_j^k - d_{ij}\right), \tag{SM32}$$

where $\overline{g}_{aj}$ is the peak conductance following a single incoming spike from the j^{th} excitatory presynaptic neuron, w_{ij} is the conductance weight defined above (1 for *E*→*I* and *I*→*E* connections, 0 for *E*→*E* and *I*→*I* connections, and *w* for connections from the input to output layer), t_j^k is the time of the k^{th} spike from neuron *j*, and d_{ij} is the synaptic delay between neurons *i* and *j*. The effect of a spike on the conductance is given by an alpha-function,

$$\alpha_\tau(t) = \frac{t}{\tau}\exp\left(1 - \frac{t}{\tau}\right), \quad \text{for } t > 0, \text{ and } 0 \text{ otherwise.} \qquad \text{(SM33)}$$

The synaptic conductance at an excitatory neuron caused by spikes from inhibitory presynaptic neurons follows an expression analogous to Equation (SM30).

The after hyperpolarizing conductance, $g_{iA}(t)$, is induced by the cell's own spikes: $g_{iA}(t) = \overline{g}_A \sum_k \alpha_{\tau_A}\left(t - t_i^k - d_A\right)$, with d_A a delay, t_i^k the time of the cell's own k^{th} spike. The leak conductance, g_L, is constant. When the membrane potential exceeds the spike threshold (-55 mV), a spike is emitted, the potential is reset to -60 mV, where it is held for an absolute refractory period. This refractory period is 3 ms for excitatory neurons and 1.5 ms for inhibitory neurons. Moreover, the spike threshold is elevated by 10 mV and exponentially decays back to -55 mV with a time constant of 10 ms; this mimics a relative refractory period.

Parameters

The reversal potentials are $E_E = 0\ \text{mV}$, $E_I = -70\ \text{mV}$, $E_A = -90\ \text{mV}$, and $E_L = -70\ \text{mV}$. The time constants for the conductances are $\tau_E = 1\ \text{ms}$ and $\tau_I = \tau_A = 2\ \text{ms}$. Excitatory neurons have $C = 0.5\ \text{mF}$, $g_L = 25\ \text{nS}$, and $\overline{g}_A = 40\ \text{nS}$. Inhibitory neurons have $C = 0.2\ \text{mF}$, $g_L = 20\ \text{nS}$, and $\overline{g}_A = 20\ \text{nS}$. The synaptic delays, d_{ij}, between inhibitory and excitatory neurons in the output layer are randomly drawn from a zero-bounded normal distribution with mean 3 ms and standard deviation 1 ms, with no delay exceeding 6 ms; the delay d_A is 1 ms. The peak conductances are given as follows: $\bar{g}_{aj} = 12$ nS if $a = E$ and j refers to a neuron in the input layer; $\bar{g}_{aj} = 10$ nS if $a = I$ and j refers to a neuron in the input layer or if $a = E$ and j refers to an inhibitory neuron in the output layer; $\bar{g}_{aj} = 3$ nS if $a = I$ and j refers to an excitatory neuron in the output layer.

For each combination of gains in the input layers we ran 1008 trials. Each trial lasted 500 ms. The equations were integrated using the Euler method with a time step of 0.5 ms.

Three networks were tested:

- σ_{input} =20, σ_{kernel} =15, and *w*=1 for both input layers;
- σ_{input} =15, σ_{kernel} =20, and *w*=1.78 for input layer 1, while σ_{input} =25, σ_{kernel} =10, and *w*=0.77 for input layer 2;
- σ_{input} =15, σ_{kernel} =15, and *w*=1.78 for input layer 1, while σ_{input} =25, σ_{kernel} =15, and *w*=0.45 for input layer 2.

Estimating the mean and variance of the posterior distribution

Ideally, one would like to use Bayes' theorem to estimate, on every trial, the mean and variance of the posterior of the distribution encoded by the excitatory neurons in the output layer. Unfortunately, this requires that we first measure the likelihood function, $p(\mathbf{r} \mid s)$. Estimating a probability distribution over 1008 neurons is in practice impossible unless the neurons are independent, which is not the case in these simulations.

Instead, we used an approach which is very similar to the one used in human experiments [1,2]. On every trial, we estimated the mean of the distribution by decoding the output pattern of activity. We then used the mean and variance of this estimate over 1008 trials as estimates of the mean and variance of the posterior distribution. This method will converge to the right values when all distributions are Gaussian and the decoder is optimal. Unfortunately, the optimal decoder also requires knowledge of $p(\mathbf{r} \mid s)$. Therefore, we used a (potentially) suboptimal decoder instead. Specifically, for the mean, we estimated the value of *s* by applying a least-squares fit of a Gaussian to the population pattern of activity on a single trial, with the amplitude, width, and peak location as parameters. (We also fit Gaussians with fixed width and amplitude and used only the peak location as a parameter; the results were the same.) The value of the peak location was used as an estimate of s. We repeated these steps over 1008 trials, and reported the estimate averaged over all trials, as is common in psychophysics. Because our decoder is not optimal, our estimates of the mean are not as good as they could be. However, we use the same estimator when only one input is active and when both are active, so a difference in optimality is expected to cancel. To estimate the variance, we used a locally optimal linear estimator [3]. We also computed the variance of the estimates themselves; these were nearly identical.

Comparing network performance to the predictions of optimal Bayesian inference

We first simulated our network with only input layer 1 active. Spike trains were generated in the input layer as described above, with a Gaussian profile centered at s_2=89.5 with gain g_1 (Fig. 3a). The gain could take any integer value between 3 and 18 spikes per second, in increments of 3 spikes/s. For a given gain we performed 1008 trials, and for each trial we measured the spike counts over 500 ms for every neuron and estimated the mean of the posterior distribution (denoted μ_1) as described above. We repeated these steps with only input layer 2 active. Spikes in the input layer followed a Gaussian profile centered at s_2=95.5 with gain g_2 (Fig. 3a). Note that we introduced a cue conflict of 6º, which is fairly large. Again we computed the mean (μ_2) of the posterior distribution encoded in the output layer. Finally, we performed simulations with both input layers active, using all combinations of gains, for a total of 36 (6×6) conditions. We used the same input spike trains as the ones generated when only one input layer was active. The output spike counts were used to compute estimates of the mean of the encoded distribution (μ_3).

After collecting all data, we computed a locally optimal linear estimator from 25,000 trials randomly chosen from all combinations of gains. The weight vector obtained in this manner was subsequently used to estimate the variances in every single condition. For each combination of gains we thus obtained estimates of σ_1^2 (only input layer 1 active), σ_2^2 (only input layer 2 active), and σ_3^2 (both input layers active). Importantly, we did not train a different estimator for every single combination of gains, but only an overall one. The intuition behind this is that the nervous system does not have the luxury of using specialized decoding circuitry for every possible contrast level of an incoming stimulus.

We then plotted μ_3 against $\mu_1 \frac{\sigma_2^2}{\sigma_1^2+\sigma_2^2} + \mu_2 \frac{\sigma_1^2}{\sigma_1^2+\sigma_2^2}$ (Equation (4), main text), and σ_3^2 against $\frac{\sigma_1^2\sigma_2^2}{\sigma_1^2+\sigma_2^2}$ (Equation (5), main text) for each combination of gains. If the network is performing a close approximation to Bayesian inference, the data should lie close to a line with slope 1 and intercept 0. This procedure was followed separately for each of three networks described above. As can be seen in Fig. 3c,d, it is clear that the network is indeed nearly optimal for all combinations of gains tested, in all three conditions.

References

1. Knill, D. C. & Richards, W. *Perception as Bayesian Inference* (Cambridge University Press, New York, 1996).
2. Ernst, M. O. & Banks, M. S. Humans integrate visual and haptic information in a statistically optimal fashion. *Nature* **415**, 429-33 (2002).
3. Series, P., Latham, P. & Pouget, A. Tuning curve sharpening for orientation selectivity: coding efficiency and the impact of correlations. *Nature Neuroscience* **10**, 1129-1135 (2004).