

A TECHNICAL APPENDICES AND SUPPLEMENTARY MATERIAL

A.1 INFORMATION GAP DERIVATION

Consider a generative model of sensory observations $\theta \rightarrow x$, where x is the noisy sensory observation (e.g. a drifting grating stimulus) generated according to the conditional distribution $p(x|\theta)$, where θ is the hidden state of the environment (e.g. true orientation of the drifting grating stimulus). Note that the likelihood function is given by $\mathcal{L}(\theta) \equiv p(x|\theta)$ for a specific observation x . Consider an experimental setup where there are two possible stimulus generation contexts: $c = \{A, B\}$ with their associated context-specific latent priors $p^c(\theta) := p(\theta|c)$ and their context frequencies $p(c)$.

Given a sensory observation x and a context c , the context-dependent posterior distribution of θ , denoted as $p^c(\theta|x) := p(\theta|x, c)$, is given by the Baye’s rule:

$$\begin{aligned} p^c(\theta|x) &= \frac{p^c(\theta, x)}{p^c(x)} \\ &= \frac{p^c(x|\theta) \cdot p^c(\theta)}{\sum_{\theta'} p^c(x|\theta') \cdot p^c(\theta')}, \quad \text{Since the generative process } \theta \rightarrow x \text{ is independent of } c, \\ &= \frac{p(x|\theta) \cdot p^c(\theta)}{\sum_{\theta'} p(x|\theta') \cdot p^c(\theta')} \\ &\propto p(x|\theta) \cdot p^c(\theta) \end{aligned}$$

For a given neural population response vector $\mathbf{r}$, consider two competing probabilistic coding hypothesis:

1. **Likelihood coding hypothesis:** $\mathbf{r}_L \sim p(x|\theta)$, where the neural population responses $\mathbf{r}_L$ is hypothesized to encode the likelihood function of the stimulus $p(x|\theta)$.
2. **Posterior coding hypothesis:** $\mathbf{r}_P \sim p(\theta|x)$, where the neural population response $\mathbf{r}_P$ is hypothesized to encode the posterior distribution of the hidden state given the stimulus $p(\theta|x)$.

We consider whether it is possible to differentiate the probabilistic information content encoded in given neural population responses $\mathbf{r}$ through a decoding approach. Intuitively, if a neural population is encoding the likelihood function, then a decoder decoding the likelihood function should lead to a better performance than a decoder decoding the posterior distribution; vice versa if the neural population is encoding the posterior distribution. In other words, decoder performance degrades when trying to decode mismatched probabilistic content, such that the difference in decoder performance when decoding the likelihood function versus decoding the posterior distribution can be used to differentiate whether a given neural population is encoding the likelihood function (likelihood coding hypothesis) or the posterior distribution (posterior coding hypothesis). Below, we formalize this intuition by deriving the expected decoder performance difference.

Consider applying a decoder function g which is optimized to decode some probabilistic information content from the neural population responses under cross-entropy loss:

$$g(\mathbf{r}) \rightarrow p(\cdot) \quad \text{where } g \text{ is a decoder function}$$

Note that to establish the expected difference between decoder performances, we assume ideal decoders in derivations. Empirically, we assume the decoder is expressive enough (e.g. a multi-layer perceptron, MLP) and fully trained, and the data is abundant such that the performance of the decoder would closely approximate that of the ideal decoder.

Adopting an information-theoretic approach, our goal is to derive the *expected difference* between decoder performances as measured in cross-entropy when decoding the likelihood function versus decoding the posterior distribution from given neural population responses $\mathbf{r}$, a quantity that we termed the *information gap*, Δ^{info} , between the two decoders under a given experimental design specified by $(p(c), p^c(\theta))$, $\forall c \in \{A, B\}$ and a generative model $p(x|\theta)$. Below we will separately derive the information gap for likelihood coding hypothesis, Δ_L^{info} , and the information gap for posterior coding hypothesis, Δ_P^{info} , respectively. As a by-product, our information-theoretic analysis framework also allows for deriving the *expected decoder performance* for each decoder under the limit of perfect decoding as measured in cross-entropy.

A.1.1 INFORMATION GAP FOR LIKELIHOOD CODING HYPOTHESIS Δ_L^{INFO}

For a likelihood coding population, the neural population responses $\mathbf{r}_L$ encode the likelihood function of the sensory stimulus, which are not modulated by and hence independent of the context prior.

$$\mathbf{r}_L \sim f(p(x|\theta)), \text{ where } f \text{ is some neural encoding function.}$$

Note that since the decoders are optimized under cross-entropy loss:

$$H(p, q) = -\mathbb{E}_p[\log q] = H(p) + D_{\text{KL}}(p || q)$$

when $q^* = p \Leftrightarrow H(p, q^*)$ is minimized.

Decoding performance of a perfect likelihood decoder g_L

Applying a likelihood decoder g_L to a likelihood coding population $\mathbf{r}_L$, we want

$$g_L(\mathbf{r}_L) \longrightarrow p(x|\theta)$$

Let us assume the observation space can be discretized into $x \in \{x_i\}$, and consider the neural population responses associated with each x_i :

$$\forall x_i, c : \mathbf{r}_{L,i}^c = \mathbf{r}_{L,i} \sim f(p(x_i|\theta))$$

Since $\mathbf{r}_{L,i}$ is context-independent, let us denote the likelihood decoder output $g_L(\mathbf{r}_{L,i}^c) = g_L(\mathbf{r}_{L,i})$. Since the ground truth context prior $p^c(\theta)$ is provided to the likelihood decoder g_L as schematized in Fig. 2 with the likelihood decoder output $g_L(\mathbf{r}_{L,i})$ and the corresponding context prior $p^c(\theta)$, the context-dependent decoded posterior distribution $q_{L,i}^c(\theta)$ is given by:

$$q_{L,i}^c(\theta) = \eta_{L,i}^c \cdot g_L(\mathbf{r}_{L,i}) \cdot p^c(\theta), \text{ where } \eta_{L,i}^c \text{ is a normalization constant.}$$

The cross-entropy loss for data samples associated with x_i, c , i.e. $H(p^c(\theta|x_i), q_{L,i}^c(\theta))$, is minimized when:

$$\begin{aligned} q_{L,i}^{c*}(\theta) &= p^c(\theta|x_i) \\ \Rightarrow \eta_{L,i}^c \cdot g_L^*(\mathbf{r}_{L,i}) \cdot p^c(\theta) &= \frac{p(x_i|\theta) \cdot p^c(\theta)}{p(x_i)} \\ \Rightarrow g_L^*(\mathbf{r}_{L,i}) &= \alpha_{L,i}^c \cdot p(x_i|\theta), \text{ where } \alpha_{L,i}^c \text{ is a constant} \end{aligned} \quad (6)$$

That is, after training, the likelihood decoder output $g_L(\mathbf{r}_{L,i})$ will converge to $g_L^*(\mathbf{r}_{L,i}) \propto p(x_i|\theta)$ given enough samples.

To get the expected cross-entropy loss across the entire data set, we marginalizing over all x_i, c , and the expected cross-entropy loss for a perfect likelihood decoder can be evaluated as:

$$\begin{aligned} \mathbb{E}_{p(x_i,c)}[H(p^c(\theta|x_i), q_{L,i}^{c*}(\theta))] &= \mathbb{E}_{p(x_i,c)}[H(p^c(\theta|x_i)) + D_{\text{KL}}(p^c(\theta|x_i) || q_{L,i}^{c*}(\theta))] \\ &= \mathbb{E}_{p(x_i,c)}[H(p^c(\theta|x_i))] \\ &= \sum_{x_i,c} H(p^c(\theta|x_i)) \cdot p(x_i, c) \\ &= \sum_{x_i,c} H(p^c(\theta|x_i)) \cdot p(c) \left[\sum_{\theta} p(x_i|\theta) p^c(\theta) \right] \\ &= \sum_{x_i} \sum_c H(p^c(\theta|x_i)) \cdot p(c) \left[\sum_{\theta} p(x_i|\theta) p^c(\theta) \right] \\ &= \sum_{x_i} \left\{ H(p^A(\theta|x_i)) \cdot p(c=A) \left[\sum_{\theta} p(x_i|\theta) p^A(\theta) \right] + \right. \\ &\quad \left. H(p^B(\theta|x_i)) \cdot p(c=B) \left[\sum_{\theta} p(x_i|\theta) p^B(\theta) \right] \right\} \end{aligned} \quad (7)$$

where the second equality holds because $D_{\text{KL}}(p^c(\theta|x_i) || q_{L,i}^{c*}(\theta)) = 0$ for a perfect likelihood decoder as derived above. That is, the expected cross-entropy loss for a perfect likelihood decoder should approach the expected posterior entropy as determined by the context frequencies and context prior distributions as given by Eq. 7.

Decoding performance of the best possible posterior decoder g_P

Applying a posterior decoder g_P to a likelihood coding population $\mathbf{r}_L$, we want:

$$g_P(\mathbf{r}_L) \longrightarrow p^c(\theta|x)$$

However, since there is no context information encoded in the population responses $\mathbf{r}_L$, the posterior decoder g_P cannot achieve the same performance as the likelihood decoder in Eq. 7, as there are scenarios where identical inputs ($\mathbf{r}_{L,i}$) are trained to map to different outputs ($p^c(\theta|x_i)$) depending on the inaccessible ground-truth context information $p^c(\theta)$.

Let us consider the neural population responses associated with each observation x_i :

$$\forall x_i, c : \mathbf{r}_{L,i}^c = \mathbf{r}_{L,i} \sim f(p(x_i|\theta))$$

The frequency of a context given the observation of data samples associated with x_i is given by:

$$\begin{aligned} p(c|x = x_i) &= \frac{p(c, x_i)}{p(x_i)} \\ &= \frac{p(c) \cdot p(x_i|c)}{\sum_{c'} p(c') \cdot p(x_i|c')} \\ &= \frac{p(c) \cdot \sum_{\theta} p^c(\theta) \cdot p(x_i|\theta)}{\sum_{c'} p(c') \sum_{\theta} p^{c'}(\theta) \cdot p(x_i|\theta)} \end{aligned}$$

Let us denote

$$\begin{aligned} S_i^A &:= p(c = A) \sum_{\theta} p^A(\theta) p(x_i|\theta) \\ S_i^B &:= p(c = B) \sum_{\theta} p^B(\theta) p(x_i|\theta) \end{aligned}$$

Hence, we can define the observation-dependent context frequency for a given x_i as:

$$\begin{aligned} \rho_i^A &:= p(c = A|x = x_i) = S_i^A / (S_i^A + S_i^B) \\ \rho_i^B &:= p(c = B|x = x_i) = S_i^B / (S_i^A + S_i^B) \end{aligned}$$

Now, let us denote the posterior decoder output $q_{P,i}(\theta) := g_P(\mathbf{r}_{L,i})$, highlighting that the output can be interpreted directly as the posterior distribution over the hidden state θ , as schematized in Fig. 2. Since the posterior decoder output is agnostic to the specific context and the associated prior $p^c(\theta)$, under cross-entropy loss, $q_{P,i}(\theta)$ is trained to minimize the expression below:

$$\begin{aligned} &\min_{q_{P,i}(\theta)} \left\{ \mathbb{E}_{p(c|x_i)} [H(p^c(\theta|x_i), q_{P,i}(\theta))] \right\} \\ &= \min_{q_{P,i}(\theta)} \left\{ \rho_i^A H(p^A(\theta|x_i), q_{P,i}(\theta)) + \rho_i^B H(p^B(\theta|x_i), q_{P,i}(\theta)) \right\} \\ &= \min_{q_{P,i}(\theta)} \left\{ - \sum_{\theta} [\rho_i^A p^A(\theta|x_i) \cdot \log q_{P,i}(\theta) + \rho_i^B p^B(\theta|x_i) \cdot \log q_{P,i}(\theta)] \right\} \\ &= \min_{q_{P,i}(\theta)} \left\{ - \sum_{\theta} [\rho_i^A p^A(\theta|x_i) + \rho_i^B p^B(\theta|x_i)] \cdot \log q_{P,i}(\theta) \right\} \end{aligned}$$

Since $p^A(\theta|x_i)$ and $p^B(\theta|x_i)$ are both probability distributions over θ , and $\rho_i^A + \rho_i^B = 1$, the expression $\rho_i^A p^A(\theta|x_i) + \rho_i^B p^B(\theta|x_i)$ represents a proper probability distribution over θ . Therefore the loss above is minimized when:

$$\begin{aligned} q_{P,i}^*(\theta) &= \rho_i^A p^A(\theta|x_i) + \rho_i^B p^B(\theta|x_i) \\ &= \frac{S_i^A}{S_i^A + S_i^B} \frac{p^A(\theta) p(x_i|\theta)}{\sum_{\theta} p^A(\theta) p(x_i|\theta)} + \frac{S_i^B}{S_i^A + S_i^B} \frac{p^B(\theta) p(x_i|\theta)}{\sum_{\theta} p^B(\theta) p(x_i|\theta)} \\ &= \frac{[p(c = A) p^A(\theta) + p(c = B) p^B(\theta)] \cdot p(x_i|\theta)}{S_i^A + S_i^B} \\ &= \frac{[p(c = A) p^A(\theta) + p(c = B) p^B(\theta)] \cdot p(x_i|\theta)}{\sum_{\theta'} \{ [p(c = A) p^A(\theta') + p(c = B) p^B(\theta')] \cdot p(x_i|\theta') \}} \end{aligned}$$

That is, after training, the best possible posterior decoder output for data samples associated with x_i , i.e. $q_{P,i}^*(\theta)$, is as if the decoder were to use a surrogate prior:

$$\tilde{p}_i(\theta) = p(c = A)p^A(\theta) + p(c = B)p^B(\theta) \quad (8)$$

which is the task-marginalized, Bayes-optimal estimator of the prior distributions over θ across contexts $c \in \{A, B\}$. Interestingly, this surrogate prior distribution is independent of x_i .

Since likelihood-coding populations $\mathbf{r}_L$ contain no prior information $p^c(\theta)$, a posterior decoder g_P trained on such population responses cannot perfectly decode the posterior distribution. Instead, the posterior decoder output converges to a Bayes-optimal estimate of context-dependent posteriors determined by the context distributions $p(c)$ and $p^c(\theta)$. To obtain the expected cross-entropy loss across the entire data set, we marginalize over all x_i, c , yielding:

$$\begin{aligned} \mathbb{E}_{p(x_i, c)}[H(p^c(\theta|x_i), q_{P,i}^*(\theta))] &= \mathbb{E}_{p(x_i, c)}[H(p^c(\theta|x_i)) + D_{\text{KL}}(p^c(\theta|x_i) \parallel q_{P,i}^*(\theta))] \\ &= \mathbb{E}_{p(x_i, c)}[H(p^c(\theta|x_i))] + \mathbb{E}_{p(x_i, c)}[D_{\text{KL}}(p^c(\theta|x_i) \parallel q_{P,i}^*(\theta))] \\ &= \text{CE loss of the perfect likelihood decoder (Eq. 7)} \\ &\quad + \mathbb{E}_{p(x_i, c)}[D_{\text{KL}}(p^c(\theta|x_i) \parallel q_{P,i}^*(\theta))] \end{aligned} \quad (9)$$

Information gap for a likelihood coding population Δ_L^{info}

From Eq. 9, let us define Δ_L^{info} , the information gap between a perfect likelihood decoder (g_L^*) and the best possible posterior decoder (g_P^*) applied on a likelihood-coding population, evaluated as the expected difference in cross-entropy loss between the two decoders:

$$\begin{aligned} \Delta_L^{\text{info}} &:= \mathbb{E}_{p(x_i, c)}[D_{\text{KL}}(p^c(\theta|x_i) \parallel q_{P,i}^*(\theta))] \\ &= \sum_{x_i, c} D_{\text{KL}}(p^c(\theta|x_i) \parallel q_{P,i}^*(\theta)) \cdot p(x_i, c) \\ &= \sum_{x_i, c} D_{\text{KL}}(p^c(\theta|x_i) \parallel q_{P,i}^*(\theta)) \cdot p(c) \left[\sum_{\theta} p(x_i|\theta)p^c(\theta) \right] \\ &= \sum_{x_i} \sum_c D_{\text{KL}}(p^c(\theta|x_i) \parallel q_{P,i}^*(\theta)) \cdot p(c) \left[\sum_{\theta} p(x_i|\theta)p^c(\theta) \right] \\ &= \sum_{x_i} \left\{ D_{\text{KL}}(p^A(\theta|x_i) \parallel q_{P,i}^*(\theta)) \cdot p(c = A) \left[\sum_{\theta} p(x_i|\theta)p^A(\theta) \right] + \right. \\ &\quad \left. D_{\text{KL}}(p^B(\theta|x_i) \parallel q_{P,i}^*(\theta)) \cdot p(c = B) \left[\sum_{\theta} p(x_i|\theta)p^B(\theta) \right] \right\} \end{aligned} \quad (10)$$

Eq. 10 provides an analytic expression for the information gap for a likelihood-coding population under a task design specified by $(p(c), p^c(\theta))$ and a generative model $p(x|\theta)$. Per observation x_i , the expression evaluates the KL divergence between the true posterior $p^c(\theta|x_i)$ and a surrogate posterior $q_{P,i}^*(\theta)$, which is the output of the best possible posterior decoder utilizing the task-marginalized, Bayes-optimal estimator of the prior distribution (Eq. 8). The KL divergence is then marginalized across x_i to derive the total expected performance difference between likelihood decoders and posterior decoders.

A.1.2 INFORMATION GAP FOR POSTERIOR CODING HYPOTHESIS Δ_P^{INFO}

For a posterior coding population, the neural population responses $\mathbf{r}_P^c$ encode the posterior distribution over θ given x under the context c , i.e. $p^c(\theta|x)$, and are therefore modulated by and dependent on the context prior $p^c(\theta)$:

$$\mathbf{r}_P^c \sim f(p^c(\theta|x)), \text{ where } f \text{ is some neural encoding function.}$$

Decoding performance of a perfect posterior decoder g_P

Applying a posterior decoder g_P to a posterior-coding population $\mathbf{r}_P$, we want

$$g_P(\mathbf{r}_P) \longrightarrow p^c(\theta|x)$$

As before, let us assume the observation space can be discretized into $x \in \{x_i\}$, and consider the neural population responses associated with each x_i :

$$\forall x_i, c : \mathbf{r}_{P,i}^c \sim f(p^c(\theta|x_i))$$

We denote the output of a posterior decoder as $q_{P,i}^c(\theta) := g_P(\mathbf{r}_{P,i}^c)$, which is context-dependent as $\mathbf{r}_{P,i}^c$ depends on the context c . As the output of the posterior decoder $q_{P,i}^c(\theta)$ can be directly interpreted as the posterior distribution (schematized in Fig. 2), the cross-entropy loss for data samples associated with x_i, c , i.e. $H(p^c(\theta|x_i), q_{P,i}^c(\theta))$, is minimized when:

$$q_{P,i}^{c*}(\theta) = p^c(\theta|x_i)$$

That is, after training, the posterior decoder output $g_P(\mathbf{r}_{P,i}^c)$ will converge to $q_{P,i}^{c*}(\theta) = p^c(\theta|x_i)$, provided sufficient training samples are available.

To obtain the expected cross-entropy loss across the entire data set, we marginalize over all x_i, c , yielding:

$$\begin{aligned} \mathbb{E}_{p(x_i, c)}[H(p^c(\theta|x_i), q_{P,i}^{c*}(\theta))] &= \mathbb{E}_{p(x_i, c)}[H(p^c(\theta|x_i)) + D_{\text{KL}}(p^c(\theta|x_i) \parallel q_{P,i}^{c*}(\theta))] \\ &= \mathbb{E}_{p(x_i, c)}[H(p^c(\theta|x_i))] \\ &= \sum_{x_i, c} H(p^c(\theta|x_i)) \cdot p(x_i, c) \\ &= \sum_{x_i, c} H(p^c(\theta|x_i)) \cdot p(c) \left[\sum_{\theta} p(x_i|\theta) p^c(\theta) \right] \\ &= \sum_{x_i} \sum_c H(p^c(\theta|x_i)) \cdot p(c) \left[\sum_{\theta} p(x_i|\theta) p^c(\theta) \right] \\ &= \sum_{x_i} \left\{ H(p^A(\theta|x_i)) \cdot p(c=A) \left[\sum_{\theta} p(x_i|\theta) p^A(\theta) \right] + \right. \\ &\quad \left. H(p^B(\theta|x_i)) \cdot p(c=B) \left[\sum_{\theta} p(x_i|\theta) p^B(\theta) \right] \right\} \quad (11) \end{aligned}$$

where the second equality holds because $D_{\text{KL}}(p^c(\theta|x_i) \parallel q_{P,i}^{c*}(\theta)) = 0$ for a perfect posterior decoder as derived above. Hence, the expected cross-entropy loss for a perfect posterior decoder on a posterior coding population should approach the expected posterior entropy as determined by the context frequencies and context prior distribution as given by Eq. 11. Note Eq. 11 is the same as the expected cross-entropy loss for a perfect likelihood decoder on a likelihood coding population as derived previously in Eq. 7.

Decoding performance of the best possible likelihood decoder g_L

Applying a likelihood decoder g_L to a posterior coding population $\mathbf{r}_P$, we want

$$g_L(\mathbf{r}_P) \longrightarrow p(x|\theta)$$

In contrast to the mismatched decoding scenario of applying a posterior decoder to a likelihood-coding population where the posterior decoder cannot perfectly decode the posterior distributions from population responses for *any* observation x_i , application of a likelihood decoder to a posterior-coding population requires more intricate considerations—we reason below that only *some* x_i would cause the likelihood decoder to fail to perfectly decode the likelihood function from a posterior-coding population. We first reiterate that the posterior population responses $\mathbf{r}_P^c$ are context dependent, which means that for the same x_i , the neural responses $\mathbf{r}_{P,i}^c$ are different across the two contexts. Hence, from the perspective of a likelihood decoder, for each x_i , the inputs (neural responses $\mathbf{r}_{P,i}^c$) are different across contexts, but the target output ($p(x_i|\theta)$) is the same. Because the ground-truth context priors $p^c(\theta)$ are explicitly provided to the likelihood decoder, this scenario “pressures”

the decoder to learn a *many-to-one* mapping, which is generally achievable for a sufficiently powerful likelihood decoder (Fig. 2C).

To identify the condition in which the likelihood decoder would fail to perfectly decode the likelihood function from posterior coding population responses, recall that when applying a posterior decoder to a likelihood-coding population, the main reason why the posterior decoder cannot be perfect is that it is forced to map identical inputs ($\mathbf{r}_{L,i}$) into multiple distinct target outputs ($p^c(\theta|x_i)$). In other words, the decoder cannot be perfect because it is trying to learn a *one-to-many* mapping. Given this insight, for the scenario of applying a likelihood decoder on a posterior-coding population, we identify the condition under which likelihood decoders are forced to map identical inputs to distinct target outputs. Consider the set of pairs $\chi := \{(x_j, x_k)\}$, where each pair (x_j, x_k) satisfies:

$$\begin{aligned} & \mathbf{r}_{P,j}^A \approx \mathbf{r}_{P,k}^B \\ \Leftrightarrow & p^A(\theta|x_j) \approx p^B(\theta|x_k), \forall \theta \quad (\text{can be measured in terms of KL divergence}) \\ \Leftrightarrow & p^A(\theta) \cdot p(x_j|\theta) \propto p^B(\theta) \cdot p(x_k|\theta), \forall \theta \end{aligned} \quad (12)$$

That is, we consider the condition $\mathbf{r}_{P,j}^A \approx \mathbf{r}_{P,k}^B$, where the inputs ($\mathbf{r}_{P,j}^A$ or $\mathbf{r}_{P,k}^B$) to the likelihood decoder g_L are (approximately) the same but the target output differs based on the context ($p(x_j|\theta)$ or $p(x_k|\theta)$). Under the assumption of ideal decoders, the set of pairs in $\chi = \{(x_j, x_k)\}$ are the only scenarios where it is impossible for an ideal likelihood decoder to be perfect. In these scenarios, identical inputs (population responses encoding the same posterior distributions) need to be decoded into different outputs (distinct likelihood functions), which is not achievable by any functional decoder, regardless of training sample size or expressive of parametrization.

With the insight that only observations in the set of pairs $\chi = (x_j, x_k)$ where Eq. 12 is satisfied will cause the likelihood decoder to fail to perfectly decode the likelihood function, let us now derive the expected likelihood decoder output for each pair. Firstly, consider the frequency of a context given an observation of neural responses associated with $\mathbf{r}_{P,j}^A$ or $\mathbf{r}_{P,k}^B$:

$$\begin{aligned} p(c = A | \mathbf{r} = \mathbf{r}_{P,j}^A \vee \mathbf{r}_{P,k}^B) &= \frac{p(c = A, \mathbf{r} = \mathbf{r}_{P,j}^A \vee \mathbf{r}_{P,k}^B)}{p(\mathbf{r} = \mathbf{r}_{P,j}^A \vee \mathbf{r}_{P,k}^B)} \\ &= \frac{p(c = A) \cdot \sum_{\theta} p^A(\theta) p(x_j|\theta)}{p(c = A) \cdot \sum_{\theta} p^A(\theta) p(x_j|\theta) + p(c = B) \cdot \sum_{\theta} p^B(\theta) p(x_k|\theta)} \end{aligned}$$

Similarly, we have:

$$p(c = B | \mathbf{r} = \mathbf{r}_{P,j}^A \vee \mathbf{r}_{P,k}^B) = \frac{p(c = B) \cdot \sum_{\theta} p^B(\theta) p(x_k|\theta)}{p(c = A) \cdot \sum_{\theta} p^A(\theta) p(x_j|\theta) + p(c = B) \cdot \sum_{\theta} p^B(\theta) p(x_k|\theta)}$$

Let us denote

$$\begin{aligned} S_j^A &:= p(c = A) \sum_{\theta} p^A(\theta) p(x_j|\theta) \\ S_k^B &:= p(c = B) \sum_{\theta} p^B(\theta) p(x_k|\theta) \end{aligned}$$

Define the observation-dependent context frequency for observing data samples coming from $\mathbf{r}_{P,j}^A$ or $\mathbf{r}_{P,k}^B$:

$$\begin{aligned} \rho_j^A &:= p(c = A | \mathbf{r} = \mathbf{r}_{P,j}^A \vee \mathbf{r}_{P,k}^B) = S_j^A / (S_j^A + S_k^B) \\ \rho_k^B &:= p(c = B | \mathbf{r} = \mathbf{r}_{P,j}^A \vee \mathbf{r}_{P,k}^B) = S_k^B / (S_j^A + S_k^B) \end{aligned}$$

Now, let us denote the context-independent likelihood decoder output as $\ell_{jk}(\theta) := g_L(\mathbf{r} = \mathbf{r}_{P,j}^A \vee \mathbf{r}_{P,k}^B)$. The context-dependent posterior distribution given the corresponding context prior $p^c(\theta)$ is

given by:

$$q_{L,j}^A(\theta) = \frac{p^A(\theta)\ell_{jk}(\theta)}{\sum_{\theta'} p^A(\theta')\ell_{jk}(\theta')} = \frac{p^A(\theta)\ell_{jk}(\theta)}{Z_j^A[\ell_{jk}(\theta)]}$$

$$q_{L,k}^B(\theta) = \frac{p^B(\theta)\ell_{jk}(\theta)}{\sum_{\theta'} p^B(\theta')\ell_{jk}(\theta')} = \frac{p^B(\theta)\ell_{jk}(\theta)}{Z_k^B[\ell_{jk}(\theta)]}$$

where $Z_j^A[\ell_{jk}(\theta)]$ and $Z_k^B[\ell_{jk}(\theta)]$ are normalization constants dependent on $\ell_{jk}(\theta)$, defined as:

$$Z_j^A[\ell_{jk}(\theta)] := \sum_{\theta} p^A(\theta)\ell_{jk}(\theta)$$

$$Z_k^B[\ell_{jk}(\theta)] := \sum_{\theta} p^B(\theta)\ell_{jk}(\theta)$$

Under cross-entropy loss, we want $\ell_{jk}(\theta)$ (and hence its associated posteriors $q_{L,j}^A(\theta)$ and $q_{L,k}^B(\theta)$) to minimize:

$$\begin{aligned} & \min_{\ell_{jk}(\theta)} \left\{ \rho_j^A H(p^A(\theta|x_j), q_{L,j}^A(\theta)) + \rho_k^B H(p^B(\theta|x_k), q_{L,k}^B(\theta)) \right\} \\ &= \min_{\ell_{jk}(\theta)} \left\{ - \sum_{\theta} \left[\rho_j^A p^A(\theta|x_j) \log q_{L,j}^A(\theta) + \rho_k^B p^B(\theta|x_k) \log q_{L,k}^B(\theta) \right] \right\} \\ &= \min_{\ell_{jk}(\theta)} \left\{ - \sum_{\theta} \left[\rho_j^A \frac{p^A(\theta)p(x_j|\theta)}{\sum_{\theta'} p^A(\theta')p(x_j|\theta')} \log \frac{p^A(\theta)\ell_{jk}(\theta)}{Z_j^A[\ell_{jk}]} + \right. \right. \\ & \quad \left. \left. \rho_k^B \frac{p^B(\theta)p(x_k|\theta)}{\sum_{\theta'} p^B(\theta')p(x_k|\theta')} \log \frac{p^B(\theta)\ell_{jk}(\theta)}{Z_k^B[\ell_{jk}]} \right] \right\} \end{aligned} \quad (13)$$

Define

$$\mu_j^A(\theta) := \rho_j^A p^A(\theta|x_j) = \rho_j^A \frac{p^A(\theta)p(x_j|\theta)}{\sum_{\theta'} p^A(\theta')p(x_j|\theta')} = \frac{p(c=A)p^A(\theta)p(x_j|\theta)}{S_j^A + S_k^B}$$

$$\mu_k^B(\theta) := \rho_k^B p^B(\theta|x_k) = \rho_k^B \frac{p^B(\theta)p(x_k|\theta)}{\sum_{\theta'} p^B(\theta')p(x_k|\theta')} = \frac{p(c=B)p^B(\theta)p(x_k|\theta)}{S_j^A + S_k^B}$$

Note

$$\sum_{\theta} \mu_j^A(\theta) = \frac{p(c=A) \sum_{\theta} p^A(\theta)p(x_j|\theta)}{S_j^A + S_k^B} = \rho_j^A$$

$$\sum_{\theta} \mu_k^B(\theta) = \frac{p(c=B) \sum_{\theta} p^B(\theta)p(x_k|\theta)}{S_j^A + S_k^B} = \rho_k^B$$

The cross-entropy loss term in Eq. 13 can be rewritten as:

$$\begin{aligned} L(\ell_{jk}(\theta)) &= - \sum_{\theta} \left[\mu_j^A(\theta) \cdot \left(\log p^A(\theta) + \log \ell_{jk}(\theta) - \log Z_j^A[\ell_{jk}(\theta)] \right) + \right. \\ & \quad \left. \mu_k^B(\theta) \cdot \left(\log p^B(\theta) + \log \ell_{jk}(\theta) - \log Z_k^B[\ell_{jk}(\theta)] \right) \right] \\ &= - \left\{ \sum_{\theta} \left[\mu_j^A(\theta) \log p^A(\theta) + \mu_k^B(\theta) \log p^B(\theta) \right] \right. \\ & \quad \left. + \sum_{\theta} \left[(\mu_j^A(\theta) + \mu_k^B(\theta)) \cdot \log \ell_{jk}(\theta) \right] \right. \\ & \quad \left. - \left[\sum_{\theta} \mu_j^A(\theta) \right] \cdot \log Z_j^A[\ell_{jk}(\theta)] - \left[\sum_{\theta} \mu_k^B(\theta) \right] \cdot \log Z_k^B[\ell_{jk}(\theta)] \right\} \end{aligned} \quad (14)$$

Note from Eq. 14, we can see that $L(\alpha\ell) = L(\ell)$, $\forall \alpha > 0$, as the normalization factors cancel out the multiplicative effect. Therefore ℓ^* that minimizes L is determined up to a multiplicative constant, agreeing with our intuition that the output of a likelihood decoder should be only determined up to a multiplicative constant as in Eq. 6.

The above minimization happens at the critical point $\ell_{jk}^*(\theta)$ where $\frac{\partial L}{\partial \ell_{jk}^*(\theta)} = 0$, $\forall \theta$, with L defined in Eq. 14.

Before proceeding to find the minimum for this variational calculus problem, let us first evaluate:

$$\begin{aligned}\frac{\partial}{\partial \ell_{jk}(\theta)} Z_j^A[\ell_{jk}(\theta)] &= \frac{\partial}{\partial \ell_{jk}(\theta)} \left\{ \sum_{\theta'} p^A(\theta') \ell_{jk}(\theta') \right\} = p^A(\theta) \\ \frac{\partial}{\partial \ell_{jk}(\theta)} Z_k^B[\ell_{jk}(\theta)] &= \frac{\partial}{\partial \ell_{jk}(\theta)} \left\{ \sum_{\theta'} p^B(\theta') \ell_{jk}(\theta') \right\} = p^B(\theta)\end{aligned}$$

To find the minimum, let us take the derivative of L with respect to $\ell_{jk}(\theta)$ and set it to zero:

$$\begin{aligned}0 &= \frac{\partial L(\ell_{jk}(\theta))}{\partial \ell_{jk}(\theta)} \\ &= -\frac{\partial}{\partial \ell_{jk}(\theta)} \left\{ \sum_{\theta} \left[\mu_j^A(\theta) \log p^A(\theta) + \mu_k^B(\theta) \log p^B(\theta) \right] \right. \\ &\quad \left. + \sum_{\theta} \left[(\mu_j^A(\theta) + \mu_k^B(\theta)) \cdot \log \ell_{jk}(\theta) \right] \right. \\ &\quad \left. - \left[\sum_{\theta} \mu_j^A(\theta) \right] \cdot \log Z_j^A[\ell_{jk}(\theta)] - \left[\sum_{\theta} \mu_k^B(\theta) \right] \cdot \log Z_k^B[\ell_{jk}(\theta)] \right\} \\ &= -\left\{ \frac{\mu_j^A(\theta) + \mu_k^B(\theta)}{\ell_{jk}(\theta)} - \frac{\left[\sum_{\theta} \mu_j^A(\theta) \right]}{Z_j^A[\ell_{jk}(\theta)]} \frac{\partial Z_j^A[\ell_{jk}(\theta)]}{\partial \ell_{jk}(\theta)} - \frac{\left[\sum_{\theta} \mu_k^B(\theta) \right]}{Z_k^B[\ell_{jk}(\theta)]} \frac{\partial Z_k^B[\ell_{jk}(\theta)]}{\partial \ell_{jk}(\theta)} \right\} \\ &= -\left\{ \frac{\mu_j^A(\theta) + \mu_k^B(\theta)}{\ell_{jk}(\theta)} - \frac{\rho_j^A}{Z_j^A[\ell_{jk}(\theta)]} p^A(\theta) - \frac{\rho_k^B}{Z_k^B[\ell_{jk}(\theta)]} p^B(\theta) \right\}\end{aligned}$$

Therefore the minimization happens when (determined up to a multiplicative constant):

$$\begin{aligned}\ell_{jk}^*(\theta) &\propto \frac{\mu_j^A(\theta) + \mu_k^B(\theta)}{\frac{\rho_j^A}{Z_j^A[\ell_{jk}^*]} p^A(\theta) + \frac{\rho_k^B}{Z_k^B[\ell_{jk}^*]} p^B(\theta)} \\ &= \frac{\rho_j^A p^A(\theta|x_j) + \rho_k^B p^B(\theta|x_k)}{\frac{\rho_j^A}{Z_j^A[\ell_{jk}^*]} p^A(\theta) + \frac{\rho_k^B}{Z_k^B[\ell_{jk}^*]} p^B(\theta)}\end{aligned}\tag{15}$$

Eq. 15 gives an implicit expression for $\ell_{jk}^*(\theta)$, since both $Z_j^A[\ell_{jk}^*]$ and $Z_k^B[\ell_{jk}^*]$ depend on $\ell_{jk}^*(\theta)$. The equation can be solved using fixed-point iteration starting with some initial guess for $\ell_{jk}^{(0)}(\theta) > 0$. For instance:

Initialize $\ell_{jk}^{(0)}(\theta) \propto 1$

for $t = 0, 1, 2, \dots$:

$$\text{compute } Z_j^{A,(t)}[\ell_{jk}^{(t)}] = \sum_{\theta} \ell_{jk}^{(t)}(\theta) p^A(\theta)$$

$$Z_k^{B,(t)}[\ell_{jk}^{(t)}] = \sum_{\theta} \ell_{jk}^{(t)}(\theta) p^B(\theta)$$

$$\text{update } \ell_{jk}^{(t+1)}(\theta) = \frac{\rho_j^A p^A(\theta|x_j) + \rho_k^B p^B(\theta|x_k)}{\frac{\rho_j^A}{Z_j^{A,(t)}[\ell_{jk}^{(t)}]} p^A(\theta) + \frac{\rho_k^B}{Z_k^{B,(t)}[\ell_{jk}^{(t)}]} p^B(\theta)}$$

Stop when $\ell_{jk}^{(t)}(\theta)$ converges (up to a multiplicative constant).

That is, as given by Eq. 15, after training, the best possible likelihood decoder output for data samples associated with $\mathbf{r}_{P,j}^A$ and $\mathbf{r}_{P,k}^B$ is as if the likelihood decoder were to divide a surrogate posterior that is a weighted sum of ground-truth posteriors $\rho_j^A p^A(\theta|x_j) + \rho_k^B p^B(\theta|x_k)$ by a surrogate prior that is a weighted sum of ground-truth priors $\frac{\rho_j^A}{Z_j^A[\ell_{jk}^*]} p^A(\theta) + \frac{\rho_k^B}{Z_k^B[\ell_{jk}^*]} p^B(\theta)$.

The posterior of the best possible likelihood decoder output $g_L^* = \ell_{jk}^*(\theta)$ given the corresponding context prior for $\mathbf{r}_{P,j}^A$ and $\mathbf{r}_{P,k}^B$ is evaluated as:

$$q_{L,j}^{A*}(\theta) = \frac{\ell_{jk}^*(\theta) p^A(\theta)}{Z_j^A[\ell_{jk}^*]}$$

$$q_{L,k}^{B*}(\theta) = \frac{\ell_{jk}^*(\theta) p^B(\theta)}{Z_k^B[\ell_{jk}^*]}$$

Hence, to obtain the expected cross-entropy loss across the entire data set, we marginalize over all x_i, c , and the total cross-entropy loss for the best possible likelihood decoder can be expressed as:

$$\begin{aligned} \mathbb{E}_{p(x_i,c)}[H(p^c(\theta|x_i), q_{L,i}^{c*}(\theta))] &= \mathbb{E}_{p(x_i,c)}[H(p^c(\theta|x_i)) + D_{\text{KL}}(p^c(\theta|x_i) \parallel q_{L,i}^{c*}(\theta))] \\ &= \mathbb{E}_{p(x_i,c)}[H(p^c(\theta|x_i))] + \mathbb{E}_{p(x_i,c)}[D_{\text{KL}}(p^c(\theta|x_i) \parallel q_{L,i}^{c*}(\theta))] \\ &= \text{CE loss for the perfect posterior decoder (Eq. 11)} \\ &\quad + \mathbb{E}_{p(x_i,c)}[D_{\text{KL}}(p^c(\theta|x_i) \parallel q_{L,i}^{c*}(\theta))] \end{aligned} \quad (16)$$

Information gap for a posterior coding population Δ_P^{info}

From equation 16, let us define Δ_P^{info} , the information gap for a posterior coding population between a perfect posterior decoder (g_P^*) and the best possible likelihood decoder (g_L^*), as the expected difference in the cross-entropy loss of the two decoders:

$$\begin{aligned} \Delta_P^{\text{info}} &:= \mathbb{E}_{p(x_i,c)}[D_{\text{KL}}(p^c(\theta|x_i) \parallel q_{L,i}^{c*}(\theta))] \\ &= \sum_{x_i,c} D_{\text{KL}}(p^c(\theta|x_i) \parallel q_{L,i}^{c*}(\theta)) \cdot p(x_i, c), \quad \text{since only } x_i \in \chi = \{(x_j, x_k)\} \text{ terms are nonzero} \\ &= \sum_{x_i \in \chi, c} D_{\text{KL}}(p^c(\theta|x_i) \parallel q_{L,i}^{c*}(\theta)) \cdot p(x_i, c) \\ &= \sum_{x_i \in \chi, c} D_{\text{KL}}(p^c(\theta|x_i) \parallel q_{L,i}^{c*}(\theta)) \cdot p(c) \left[\sum_{\theta} p(x_i|\theta) p^c(\theta) \right] \\ &= \sum_{x_i \in \chi} \sum_c D_{\text{KL}}(p^c(\theta|x_i) \parallel q_{L,i}^{c*}(\theta)) \cdot p(c) \left[\sum_{\theta} p(x_i|\theta) p^c(\theta) \right] \\ &= \sum_{(x_j, x_k)} \left\{ D_{\text{KL}}(p^A(\theta|x_j) \parallel q_{L,j}^{A*}(\theta)) \cdot p(c=A) \left[\sum_{\theta} p(x_j|\theta) p^A(\theta) \right] \right. \\ &\quad \left. + D_{\text{KL}}(p^B(\theta|x_k) \parallel q_{L,k}^{B*}(\theta)) \cdot p(c=B) \left[\sum_{\theta} p(x_k|\theta) p^B(\theta) \right] \right\} \end{aligned} \quad (17)$$

Eq. 17 provides an analytic expression for the information gap for a posterior-coding population under a task design specified by $(p(c), p^c(\theta))$ and a generative model $p(x|\theta)$. Per pair of observations (x_j, x_k) , we evaluate the KL divergence between the true posterior ($p^A(\theta|x_j)$ or $p^B(\theta|x_k)$) and a surrogate posterior ($q_{L,j}^{A*}(\theta)$ or $q_{L,k}^{B*}(\theta)$), which is the posterior distribution associated with the output of the best possible likelihood decoder utilizing the task-marginalized, Bayes-optimal estimators as given by Eq. 15. The KL divergence is then marginalized across the pairs (x_j, x_k) to derive the total expected performance difference between likelihood decoders and posterior decoders.

A.2 SCHEMATICS FOR TASK DESIGN TRADEOFF

Q: What task design would maximize Δ_L^{info} & Δ_P^{info}

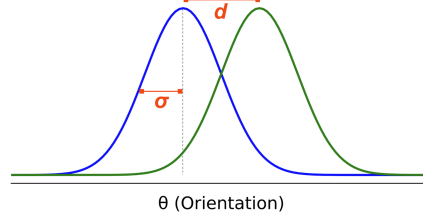


Figure 8: **Design tradeoff.**

A.3 DETAILS OF SIMULATION EXPERIMENT

A.3.1 SIMULATED NEURAL POPULATIONS

We consider tasks with Gaussian context priors motivated by classic orientation discrimination psychophysical tasks (Orbán et al., 2016; Walker et al., 2020). In this task, subjects perform an orientation discrimination under two contexts $c \in \{A, B\}$, with the context for each session sampled randomly, i.e. $p(c = A) = p(c = B) = 0.5$. Within each session, the trial-to-trial hidden world state θ (i.e. orientation) is drawn from context-specific Gaussian prior distributions $p^c(\theta) = \mathcal{N}(\mu^c, (\sigma^c)^2)$, where μ^c and $(\sigma^c)^2$ are task-specific parameters. In the simulation, we consider $\theta \in \{-90^\circ, 90^\circ\}$, and use identical variances for the two Gaussian priors $\sigma^A = \sigma^B = \sigma$. Consequently, the experimental design is fully specified by the tuple of task parameters (μ^A, μ^B, σ) . Furthermore, foregoing cardinal orientation consideration, the circular symmetry of orientations θ suggests that only the separation between the two means $d = |\mu^A - \mu^B|$ would meaningfully impact perception. Given this, we always center the two means around zero, meaning $\mu^A = -\frac{1}{2}d$ and $\mu^B = \frac{1}{2}d$. We systematically vary (d, σ) to cover the task spectrum of Gaussian context priors in the simulation studies.

Noisy sensory observations x are drawn from the conditional distribution defined by the given generative model $p(x|\theta)$. This stochastic process can be seen as capturing both intrinsic neuronal noise and uncertainty in the extrinsic stimulus features. This generative model can be experimentally manipulated through stimulus parameters such as contrast, where lower contrast induces increased observation variance, reflecting increased sensory uncertainty. In the simulation, $p(x|\theta)$ is modeled as Gaussian distributions to reflect Gaussian orientation tuning curves commonly found among simple V1 neurons. We model the effect of different contrast levels by systematically varying the standard deviation of the generative model σ_{obs} (Walker et al., 2020). To this end, standard deviations σ_{obs} of 8, 15, and 25 are chosen to model the generative model under high, medium, and low contrast levels, respectively. Finally, on each trial, the hidden world state θ is drawn from $p^c(\theta) = \mathcal{N}(\mu^c, (\sigma^c)^2)$ and then the observation is drawn from the conditional distribution $p(x|\theta) = \mathcal{N}(\theta, \sigma_{\text{obs}}^2)$.

For simulated population responses, we first implement Poisson neuron models with Gaussian tuning curves and Poisson variability (Walker et al., 2020). A population of neurons indexed by l , ranging from 5-500 neurons, was constructed with Gaussian tuning curves $\mathcal{N}(\theta_l, \sigma_{\text{obs}}^2)$ with their means θ_l tiling up the orientation space and their standard deviations being σ_{obs} . For likelihood-coding populations, the mean firing rate of each neuron on each trial, after an observation x is sampled, is

determined by the probability density of its Gaussian tuning curve, i.e. $f(x) = \frac{1}{\sqrt{2\pi\sigma_{\text{obs}}^2}} e^{-\frac{(x-\theta_l)^2}{2\sigma_{\text{obs}}^2}}$,

scaled with a fixed constant of 30 to approximate the typical range of neuron firing rates observed experimentally (Walker et al., 2020). For posterior-coding populations, the mean firing rate of each neuron is further multiplied by the context-specific prior $p^c(\theta)$, thus effectively encoding the posterior $p^c(\theta|x) \propto p(x|\theta) \cdot p^c(\theta)$ in their mean firing rates. For both populations, trial-to-trial spike counts are then generated by sampling from Poisson distribution with the specified mean firing rates.

For some simulation experiments, we additionally implemented a more complex, gain-modulated Poisson neuron model for simulating population responses (Goris et al., 2014). The gain-modulated Poisson neuron model has been proposed to account for the supra-Poisson variability commonly observed experimentally among V1 neurons. In this model, the mean firing rate of the neuron is

the product of two terms: 1) the original rate determined by the Gaussian tuning curve model, and 2) a stimulus-independent gain factor G . Goris et al. (2014) proposed and validated on V1 neural data that this stimulus-independent gain factor G can be effectively modeled as following a gamma distribution with a mean of one and variance of σ_G^2 . Based on their results, we choose a biologically realistic value of $\sigma_G \approx 0.5$ in our simulation. Therefore, on a trial-to-trial basis, after an original rate is determined according to the procedure in the previous paragraph for the likelihood-coding or posterior-coding population, a random gain factor is then sampled from the gamma distribution and multiplied with the original rate to get the mean firing rate for the gain-modulated Poisson neuron model. Similarly, spike counts are then generated from the mean firing rates with Poisson variability.

A.3.2 EXAMPLE LIKELIHOOD AND POSTERIOR

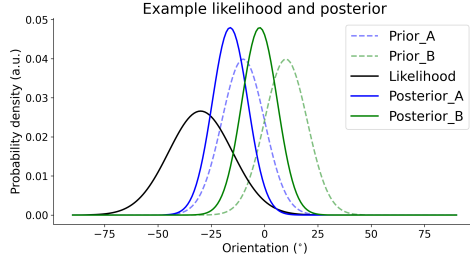


Figure 9: Example of the ground truth likelihood, priors, and posteriors.

A.3.3 PROBABILISTIC INFORMATION DECODER

As described in Fig. 2C, deep neural networks parametrized by multi-layered perceptrons are trained with cross-entropy loss to serve as flexible, powerful decoders to decode either the likelihood function or the posterior distribution from simulated neural population responses (Walker et al., 2020). We use fully-connected, deep neural networks with two hidden layers, with 300 and 200 units in the first and second layer, respectively. All hidden units are rectified linear units, and dropout rates of 0.5 are used for both layers. The input dimension to the first layer is the number of neurons in the simulated population, ranging from 5–500. The output layer is a fully connected readout with no nonlinearity and a dimension of the number of possible hidden states. In our simulation, we consider orientation $\theta \in \{-90^\circ, 90^\circ\}$ and discretize them into one degree bins, leading to a total number of possible hidden states of 181. To facilitate numerical stability, the decoded probability quantity is operating in the log space. The posterior decoder output is treated as the log-posterior, which is directly optimized to minimize the cross-entropy loss. The likelihood decoder output is treated as the log-likelihood, which is then integrated with the ground truth log-prior to arrive at the final output that is optimized to minimize the cross-entropy loss. To encourage smoothness of the decoded probability distributions, an L_2 regularizer on the log-posteriors filtered with a Laplacian filter of the form $h = [0.25, 0.5, 0.25]$ is added to the cross-entropy term, as proposed in (Walker et al., 2020). We use (0.8, 0.2) for train-validation split for training the decoders. A held-out test set is used for final evaluation and all results in the paper are on the test set. Early stop with patience of 10 and minimal change of $2e-6$ in validation set cross-entropy loss is adopted to prevent overfitting. All models were constructed and trained using the Pytorch framework (Paszke et al., 2019).

A.4 INCORPORATING BIASED PRIOR FROM BEHAVIOR DATA

Although our main analysis focuses on the theoretical decoding limit using the optimal priors, our framework naturally accommodates model mismatch or biased priors by incorporating behavioral (psychometric) measurements. The procedure is detailed in Fig. 10:

- **Analyze the psychometric curve:** In perceptual tasks (e.g., orientation discrimination tasks), deviations in the subject’s psychometric curve (correct rate as a function of stimulus orientation) from the ideal observer reveal model mismatch or biased priors.
- **Infer the subject’s model mismatch/ biased prior:** Features (such as leftward shifts or increased slope/variance) in the psychometric function can be mapped to corresponding biases or increased uncertainty in the subject’s internal mismatched prior.
- **Compute the information gap using the inferred prior:** The inferred biased prior can then be used directly in our information-gap calculation, yielding predictions that account for the subject’s model mismatch and more accurately reflect expected empirical decoder differences.

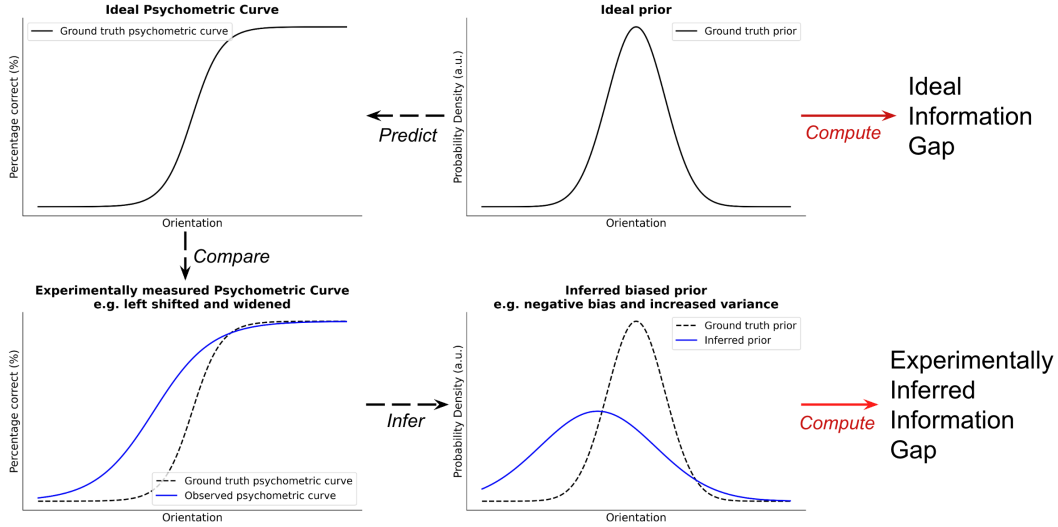


Figure 10: The information gap computation can incorporate behavior data by estimating the subject’s biased prior from its psychometric curve.

A.5 EXTENDING OUR FRAMEWORK TO MIXED CODING HYPOTHESIS

In this section we discuss a more nuanced probabilistic coding hypothesis and how our proposed framework could be extended to identify or falsify it. As an example of *mixed* coding hypothesis, Ganguli & Simoncelli (2010; 2014) proposed combining heterogeneous tuning curves—which embed aspects of the prior—with spiking variability that reflects the likelihood, yielding a hybrid code in which sensory responses carry both likelihood information and a structurally instantiated prior. This example can be categorized as a mixed or intermediate hypothesis, in between the canonical pure likelihood and pure posterior coding hypothesis. Our framework can naturally accommodate such mixed coding hypothesis by evaluating how each decoder performs under mismatched information. As shown in Fig. 11 since now both the likelihood and posterior decoders can recover the correct distributions, our theory predicts an information gap $\Delta_{\text{info}} = 0$. This zero-info-gap signature is distinct and does not arise under optimized task designs for either pure likelihood- or pure posterior-coding populations, which produce reliably nonzero and separable values. As a result, optimizing the task to maximally separate the two canonical hypotheses simultaneously maximizes sensitivity to departures from them. A mixed code that yields $\Delta_{\text{info}} = 0$ under the same optimized design becomes cleanly identifiable as neither pure likelihood nor pure posterior. Thus this discussion illustrates how our method could generalize beyond the two extreme hypotheses and provides a principled tool for distinguishing both pure and mixed coding schemes.

More broadly, we do not claim that likelihood and posterior coding are the only relevant theories in the literature, but they represent the two major families of theories that differ in what probabilistic quantity is encoded. Our contribution is to provide a principled methodology for experimentally distinguishing such theories. By optimizing task parameters to maximally separate these canonical extremes, we simultaneously maximize sensitivity to discriminating more nuanced probabilistic coding theories like the mixed coding hypothesis.

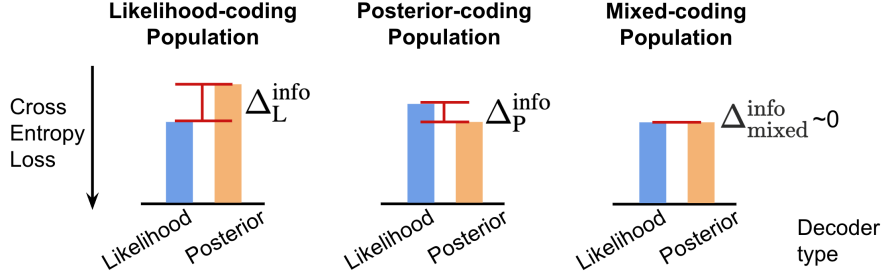


Figure 11: Under mixed coding hypothesis, the information gap becomes zero.

A.6 EFFECT OF NOISE AND FIRING RATE

As shown in Fig. 12, decreasing firing rates or increasing noise slows the convergence of empirical decoder performance differences. More trials are needed for the empirical decoder performance difference to approach the theoretical information gap. However, with sufficient data, the decoder performance differences ultimately converge to the same theoretical value. This reflects the expected effect of reduced signal-to-noise ratio—decoding becomes harder, but the underlying difference in decodable information is unchanged. Thus, while low SNR increases data requirements, the theoretical information gap remains the correct predictor of the asymptotic difference between the two hypotheses.

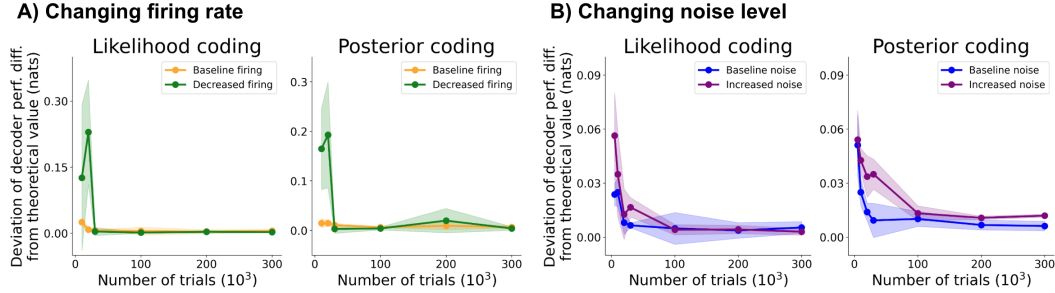


Figure 12: Effect of firing rate and noise level.

A.7 FACTORS AFFECTING CONVERGENCE SPEED

We conducted ablation studies to examine the factors that determine how quickly empirical performance converges to the theoretical value of information gap. Our main simulations assume that neural tuning curves tile the full orientation space, consistent with standard V1 models (Rubin et al., 2015). In Fig. 13A, when the population is randomly sampled without full coverage of the entire orientation space, since no decoder can recover information about orientations lacking tuned neurons, we found that convergence with respect to neuron count becomes substantially slower. In addition, in Fig. 3, the neuron-scaling experiment uses 30k trials so that decoders quickly approach the theoretical limit. In Fig. 13B, we performed an ablation with fewer trials (3k trials) and observed that convergence is again slower because the decoder cannot reliably estimate the encoded distributions from limited data. In practice, the above factors can be mitigated by modern neurophysiology population recordings that provide large number of trials with hundreds to thousands of simultaneously recorded neurons that cover full range of orientation space.

Finally, to demonstrate that our result is robust to the level of discretization of the orientation variable, we repeated the convergence analysis with higher-resolution orientation bins (0.25° instead of 1° as in the main results), and obtained indistinguishable results. This confirms that the accuracy of information gap and its empirical convergence are robust to binning and reflect the underlying decodable information rather than numerical artifacts.

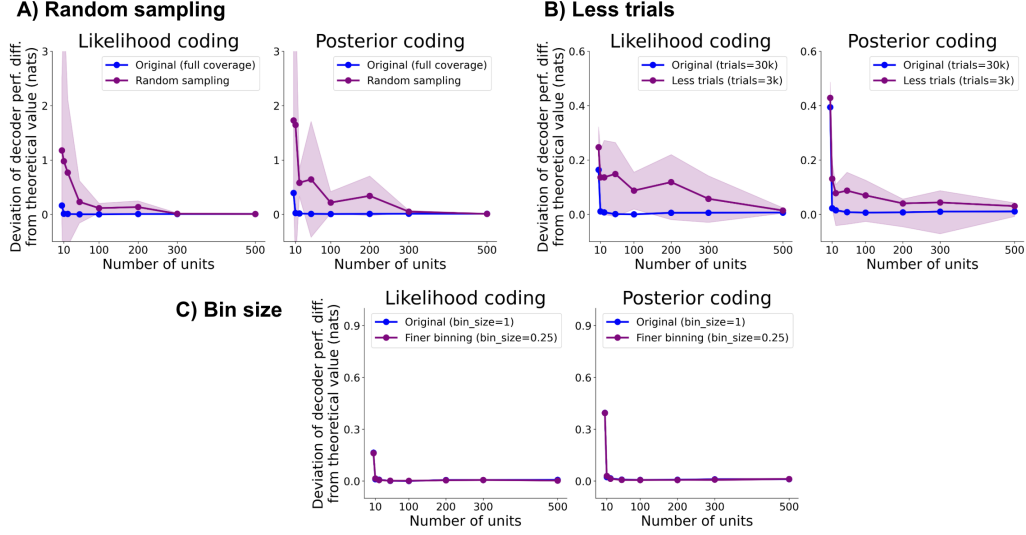


Figure 13: Examine factors affecting convergence speed. A) Effect of orientation coverage. B) Effect of trial numbers. C) Effect of bin size.

A.8 DETAILED RESULTS ON HEAVY-TAILED PRIORS

Below we first provide the full results of information gap landscape across various contrast levels for heavy-tailed context priors including student's t-distribution (Fig. 14) and Cauchy distribution (Fig. 15). Note that for t-distribution we report the results using degrees of freedom $\nu = 3$. When $\nu \rightarrow \infty$ the t-distribution reduces to a standard Gaussian distribution, and when $\nu = 0$ the t-distribution becomes the Cauchy distribution. We then provide an intuitive example explaining why the information gap for posterior coding hypothesis is dramatically lower under heavy-tailed context priors compared to Gaussian context priors. The main reason is that under Gaussian generative models, when integrated with heavy-tailed priors, the posteriors tend to become asymmetric (as opposed to Gaussian priors where the posteriors are still symmetric Gaussian), thus limiting the number of pairs (x_j, x_k) that could confuse the likelihood decoder.

A.8.1 INFORMATION GAP LANDSCAPE

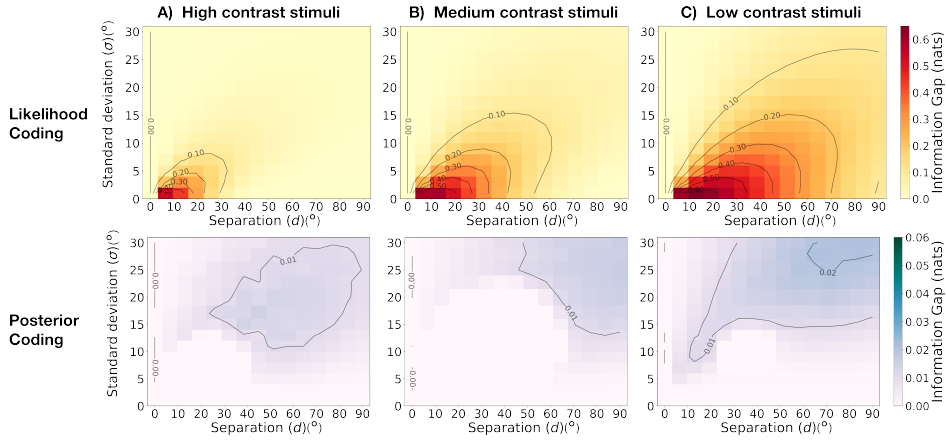


Figure 14: **Information gap landscapes when using student’s t-distribution with degrees of freedom $\nu = 3$ as context priors.** A) Information gap as a function of task parameters (d : separation between context priors, and σ : context prior standard deviations) for both the likelihood coding hypothesis (top) and the posterior coding hypothesis (bottom) when presented with high contrast stimuli. B) Same for medium contrast stimuli and C) for low contrast stimuli.

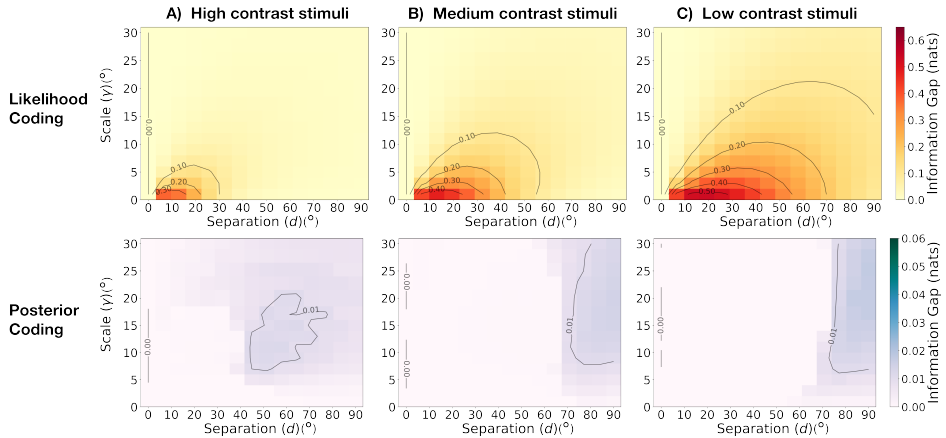


Figure 15: **Information gap landscapes when using Cauchy distribution as context priors.** A) Information gap as a function of task parameters (d : separation between context priors, and γ : context prior scales) for both the likelihood coding hypothesis (top) and the posterior coding hypothesis (bottom) when presented with high contrast stimuli. B) Same for medium and C) low contrast stimuli.

A.8.2 AN EXAMPLE EXPLAINING WHY Δ_p^{INFO} IS DRAMATICALLY DECREASED UNDER HEAVY-TAILED CONTEXT PRIORS

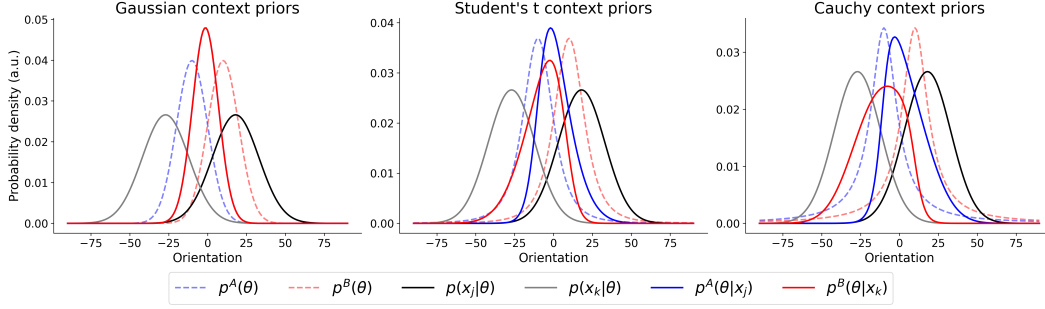


Figure 16: **Heavy tailed context priors, when integrated with Gaussian likelihood function, lead to asymmetric posterior distributions, limiting the pairs of identical posteriors satisfying Eq. 12 that would cause imperfect likelihood decoders on posterior-coding populations.** Across task designs with Gaussian context priors (left), student’s t context priors with $\nu = 3$ (middle), and Cauchy context priors (right), the context priors $p^A(\theta)$ and $p^B(\theta)$ are shown in dashed blue and red lines, respectively. Note they all share identical standard deviation or scale parameters to facilitate comparison. One example pair of $(x_j, x_k) = (18^\circ, -27^\circ)$ that satisfies Eq. 12 under Gaussian context priors is shown here, with the associated likelihood functions $p(x_j|\theta)$ and $p(x_k|\theta)$ plotted in solid gray and black lines, respectively. The posterior distributions under each context priors, $p^A(\theta|x_j)$ and $p^B(\theta|x_k)$ are shown as solid blue and red lines, respectively. Under Gaussian context priors (left), the two posteriors are equal to each other, i.e. $p^A(\theta|x_j) = p^B(\theta|x_k)$, hence the two lines overlap. However, as the context priors become increasingly heavy-tailed as under student’s t distribution (middle) and Cauchy distributions (right), the two posteriors become more and more asymmetric, leading to non-identical posterior distributions that no longer satisfy Eq. 12. This example demonstrates why there are much less pairs (x_j, x_k) that would satisfy Eq. 12 accounting for the observation that the information gap of posterior-coding population is dramatically decreased under heavy-tailed context priors.

A.9 RESULTS ON THIN-TAILED PRIORS

To provide further examples on non-Gaussian context priors, we examined thin-tailed distributions as stimulus prior distributions. We reported additional analyses using canonical thin-tailed generalized normal distributions with $\beta > 2$) in Fig. 17. The information gap landscape shows that thin-tailed priors similarly lead to near-0 posterior-coding information gaps across task parameter space. Our framework provides a similar explanation: under thin-tailed context priors, the resulting posteriors become highly asymmetric across contexts (Fig. 18), reducing the feasible set of (x_j, x_k) pairs that can satisfy Eq. 12, thereby shrinking the posterior-coding information gap, which mirrors the failure mode observed with heavy-tailed priors.

What about uniform priors? As shown in Fig. 18, a uniform prior induces no context-dependent modulation of the posterior. Hence, likelihood- and posterior-coding populations become nearly indistinguishable, causing the information gap to collapse for both hypotheses.

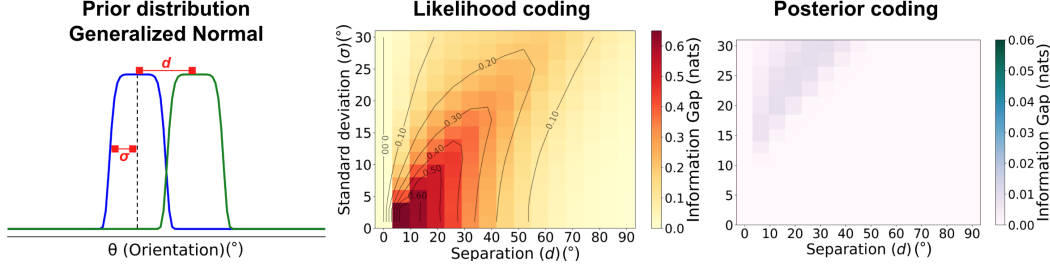


Figure 17: **Information gap landscapes when using generalized normal distribution as context priors.** A) Information gap as a function of task parameters (d : separation between context priors, and σ : context prior standard deviations) for both the likelihood coding hypothesis (middle) and the posterior coding hypothesis (right).

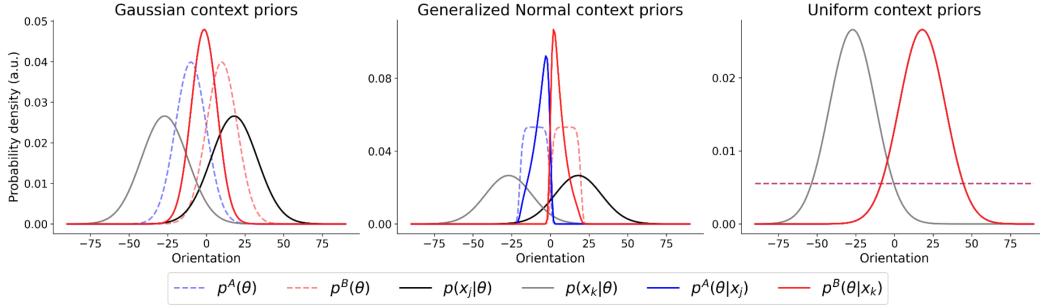


Figure 18: Thin tailed context priors, when integrated with Gaussian likelihood function, lead to asymmetric posterior distributions, limiting the pairs of identical posteriors satisfying Eq. 12 that would cause imperfect likelihood decoders on posterior-coding populations.