Learning neural codes for perceptual uncertainty

Mehrdad Salmasi*, Maneesh Sahani*
*Gatsby Computational Neuroscience Unit, University College London, United Kingdom
Emails: m.salmasi@ucl.ac.uk, maneesh@gatsby.ucl.ac.uk

Abstract—**Perception is an inferential process, in which the state of the immediate environment must be estimated from sensory input. Inference in the face of noise and ambiguity requires reasoning with uncertainty, and much animal behaviour appears close to Bayes optimal. This observation has inspired hypotheses for how the activity of neurons in the brain might represent the distributional beliefs necessary to implement explicit Bayesian computation. While previous work has focused on the sufficiency of these hypothesised codes for computation, relatively little consideration has been given to optimality in the representation itself. Here, we adopt an encoder-decoder approach to study representational optimisation within one hypothesised belief encoding framework: the distributed distributional code (DDC). We consider a setting in which typical belief distribution functions take the form of a sparse combination of an underlying set of basis functions, and the corresponding DDC signals are corrupted by neural variability. We estimate the conditional entropy over beliefs induced by these DDC signals using an appropriate decoder. Like other hypothesised frameworks, a DDC representation of a belief depends on a set of fixed encoding functions that are usually set arbitrarily. Our approach allows us to seek the encoding functions that minimise the decoder conditional entropy and thus optimise representational accuracy in an information theoretic sense. We apply the approach to show how optimal encoding properties may adapt to represent beliefs in new environments, relating the results to experimentally reported neural responses.**

I. Introduction

The brain receives information about the external world through noisy sensory signals, from which it must derive inferences about behaviourally relevant variables. Measurements of behaviour suggest that when these relevant variables are not fully constrained by sensory input, neural computation reflects processing of the full distributional belief in accordance with the principles of Bayesian inference. This observation has led to the Bayesian coding hypothesis, postulating that these distributional beliefs (or posterior probabilities) are encoded explicitly in the activity of neural populations that form the basis of perceptual inference and learning [1]. A number of frameworks have been suggested for the representation of probabilities in the brain, including probabilistic population coding [2], [3], sampling [4], [5], and distributed distributional coding (DDC) [6], [7].

In the DDC framework, a probability density function is represented by the expected values of a set of encoding functions. If the set of encoding functions is rich enough, the DDC expectations provide sufficient information to carry out probabilistic computation. Previous studies have shown that DDC representations provide an effective substrate to learn and to make inferences in deep generative models [8], to build successor representations within partially observable Markov decision processes [9], and to carry and resolve uncertainty over time [10]. In these studies, the set of encoding functions was chosen arbitrarily, and remained fixed throughout. Little is known about how the DDC encoding might be optimised so as to maximise the fidelity of the resulting distributional representations.

Here, we tackle this question using an information-theoretic learning framework. The optimal set of encoding functions will depend on assumptions, including the form of the belief distributions to be encoded, noise in the representation, and the objectives of downstream processing. We consider a setting in which the distribution functions of the beliefs to be encoded take the form of a sparse combination of an underlying set of basis functions (with positive or negative weights). These beliefs are then represented by the expectations of encoding functions as defined by the DDC, but these encoded values are corrupted by independent internal neural noise before further processing. Finally, in place of downstream processing we consider a computational decoder, which seeks to recover the encoded belief given the DDC expected values and knowledge of the sparse family from which the belief is drawn.

These assumptions allow us to exploit the generative model used in Bayesian compressive sensing and relevance vector machines. Using this framework, we derive the posterior distribution over the belief distribution function obtained by the decoder, and calculate the conditional entropy. We suggest that appropriate DDC encoding functions are those that minimise this uncertainty about the encoded distribution function. This approach provides us with a general rule for learning the encoding functions; descending the gradient of the conditional entropy over the distribution function. We show that our suggested optimality criterion can explain the dynamics of the neuronal tuning functions, providing evidence for the codes of uncertainty in the brain.

II. A generative model for DDC

The moments of a probability distribution function can provide substantial information about the characteristics of the underlying distribution. Distributed distributional coding (DDC) extends the notion of moments and suggests that the distribution over a latent variable Z is represented in the brain by the expected values of a given set of encoding functions, $\{\phi_i(z)\}_{i=1}^{K}$ [6], [7]. The distribution function $\gamma(z)$ is then represented by K DDC values,

$$r_i = \int_z \gamma(z)\phi_i(z)dz, \quad i = 1, 2, ..., K \tag{1}$$

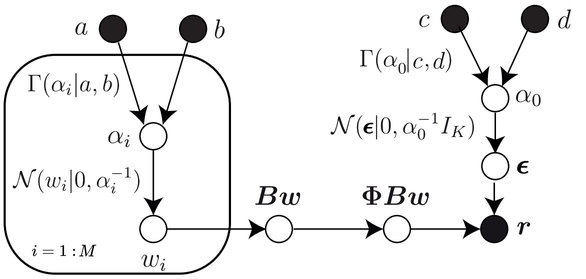

Fig. 1: The generative model for DDC values, $\boldsymbol{r}$. The distribution function, γ, has a sparse representation in some basis, $\boldsymbol{B}$, i.e. $\gamma = \boldsymbol{Bw}$, where w is the vector of sparse coefficients. The observed DDC values are generated by adding noise, $\boldsymbol{\epsilon}$, to the expected values of DDC encoding functions with respect to the distribution function, $\boldsymbol{\Phi Bw}$. The sparseness of $\boldsymbol{w}$ is fulfilled by a hierarchical prior.

Here, we assume that the latent variable, Z, is discrete with S states, and denote the distribution function by the vector $\boldsymbol{\gamma}$ and the encoding functions by the $K \times S$ matrix $\boldsymbol{\Phi}$. Therefore, in the absence of noise, the vector of DDC values, $\boldsymbol{r}$, is derived from $\boldsymbol{r} = \boldsymbol{\Phi\gamma}$. We assume that the posterior distribution $\boldsymbol{\gamma}$ has a sparse representation in some basis set of size M. The basis matrix is represented by the $S \times M$ matrix $\boldsymbol{B}$, and $\boldsymbol{\gamma} = \boldsymbol{Bw}$, where $\boldsymbol{w}$ is a vector of size M and contains the sparse coefficients of $\boldsymbol{\gamma}$. By defining $\boldsymbol{\Psi} = \boldsymbol{\Phi B}$, we have $\boldsymbol{r} = \boldsymbol{\Phi Bw} = \boldsymbol{\Psi w}$.

We model the DDC values by a generative model (Fig. 1) similar to the ones used in relevance vector machine [11], [12] and Bayesian compressive sensing [13], [14]. In this model, the sparseness criterion is implemented by a hierarchical prior for $\boldsymbol{w}$. We consider a Gaussian prior for $\boldsymbol{w}$,

$$P(\boldsymbol{w}|\boldsymbol{\alpha}) = \prod_{i=1}^{M} \mathcal{N}(w_i|0,\alpha_i^{-1}) \tag{2}$$

where α_i is the precision of the Gaussian density function. We also assume a gamma prior for $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, ..., \alpha_M)$,

$$P(\boldsymbol{\alpha}) = \prod_{i=1}^{M} \Gamma(\alpha_i|a,b) \tag{3}$$

By marginalizing out α_i in this hierarchical prior,

$$P(w_i) = \int_0^\infty \mathcal{N}(w_i|0,\alpha_i^{-1})\Gamma(\alpha_i|a,b)d\alpha_i, \tag{4}$$

and it can be shown that w_i has a t-distribution [12], [13]. In practice, a and b are set to small values (e.g. $a = b = 10^{-4}$), and the t-distribution will induce the desired sparseness for the values of w_i.

Neuronal activity in the brain is noisy. The DDC values, $\boldsymbol{r}$, are modeled by adding i.i.d. noise to the expected values of the encoding functions, $\boldsymbol{r} = \boldsymbol{\Phi Bw} + \boldsymbol{\epsilon}$, where each entry of $\boldsymbol{\epsilon}$ has a Gaussian distribution with precision α_0. We can similarly assume a gamma prior for α_0 (with parameters c and d), and set the parameters to a small value (e.g. 10^{-4}). Let

$$D_{\boldsymbol{\alpha}} = \mathrm{diag}(\alpha_1, \alpha_2, \ldots, \alpha_M), \tag{5}$$

be the diagonal matrix of the precision values. Since

$$P(\boldsymbol{w}|\boldsymbol{r},\boldsymbol{\alpha}) = \frac{P(\boldsymbol{r}|\boldsymbol{w})P(\boldsymbol{w}|\boldsymbol{\alpha})}{P(\boldsymbol{r}|\boldsymbol{\alpha})}, \tag{6}$$

it can be shown that [13],

$$P(\boldsymbol{w}|\boldsymbol{r},\boldsymbol{\alpha}) = \mathcal{N}(\boldsymbol{\mu_w}, \Sigma_{\boldsymbol{w}}) \tag{7}$$

with

$$\boldsymbol{\mu_w} = \alpha_0 \Sigma_{\boldsymbol{w}} \boldsymbol{\Psi}^T \boldsymbol{r} \tag{8}$$

$$\Sigma_{\boldsymbol{w}} = (\alpha_0 \boldsymbol{\Psi}^T \boldsymbol{\Psi} + D_{\boldsymbol{\alpha}})^{-1} \tag{9}$$

The probability of evidence given the precision values is

$$P(\boldsymbol{r}|\boldsymbol{\alpha},\alpha_0) = \int_{\boldsymbol{w}} P(\boldsymbol{r}|\boldsymbol{w},\alpha_0)P(\boldsymbol{w}|\boldsymbol{\alpha})d\boldsymbol{w}. \tag{10}$$

We can find the optimal value of $\boldsymbol{\alpha}$ using type II maximum likelihood [12], [15],

$$L(\boldsymbol{\alpha},\alpha_0) = \log P(\boldsymbol{r}|\boldsymbol{\alpha},\alpha_0) \tag{11}$$

$$= \frac{-1}{2}\left(K\log(2\pi) + \log|\Lambda| + \boldsymbol{r}^T\Lambda^{-1}\boldsymbol{r}\right) \tag{12}$$

where $\Lambda = \alpha_0^{-1}\boldsymbol{I} + \boldsymbol{\Psi} D_{\boldsymbol{\alpha}}^{-1} \boldsymbol{\Psi}^T$, and $\boldsymbol{\alpha}^* = \arg\max_{\boldsymbol{\alpha}} L(\boldsymbol{\alpha})$. An iterative algorithm has been suggested for calculating $\boldsymbol{\alpha}^*$ [13]. For a given $\boldsymbol{\mu_w}$ and $\Sigma_{\boldsymbol{w}}$, an update equation is derived for $\boldsymbol{\alpha}$ by calculating the derivative of (12) with respect to $\boldsymbol{\alpha}$,

$$\alpha_i^{\text{new}} = \frac{1 - \alpha_i \Sigma_{\boldsymbol{w}}(i,i)}{\boldsymbol{\mu}_{\boldsymbol{w}}^2(i)}, \tag{13}$$

and we iterate over (5), (8), (9), and (13) until convergence. Similarly, we can find the optimal value of α_0, however in this study, we assume that the neuronal noise variance is known.

For the optimal value $\boldsymbol{\alpha}^*$, $P(\boldsymbol{w}|\boldsymbol{r},\boldsymbol{\alpha}^*) = \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{w}}^*, \Sigma_{\boldsymbol{w}}^*)$, with

$$\boldsymbol{\mu}_{\boldsymbol{w}}^* = \alpha_0 \Sigma_{\boldsymbol{w}}^* \boldsymbol{\Psi}^T \boldsymbol{r} \tag{14}$$

$$\Sigma_{\boldsymbol{w}}^* = (\alpha_0 \boldsymbol{\Psi}^T \boldsymbol{\Psi} + D_{\boldsymbol{\alpha}^*})^{-1} \tag{15}$$

Since $\boldsymbol{\gamma} = \boldsymbol{Bw}$, we can easily find the posterior of $\boldsymbol{\gamma}$,

$$P(\boldsymbol{\gamma}|\boldsymbol{r},\boldsymbol{\alpha}^*) = \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{\gamma}}^*, \Sigma_{\boldsymbol{\gamma}}^*) \tag{16}$$

with

$$\boldsymbol{\mu}_{\boldsymbol{\gamma}}^* = \boldsymbol{B}\boldsymbol{\mu}_{\boldsymbol{w}}^* \tag{17}$$

$$\Sigma_{\boldsymbol{\gamma}}^* = \boldsymbol{B}\Sigma_{\boldsymbol{w}}^*\boldsymbol{B}^T \tag{18}$$

III. Learning DDC encoding functions

We hypothesize that the DDC encoding functions are modified to reduce the amount of uncertainty about the distribution function $\boldsymbol{\gamma}$ given the DDC values, $\boldsymbol{r}$. The uncertainty is quantified by the conditional differential entropy $h(\boldsymbol{\gamma}|\boldsymbol{r})$, however, since we are using type II maximum likelihood for estimating the precision parameters, we calculate instead, $h(\boldsymbol{\gamma}|\boldsymbol{r},\boldsymbol{\alpha}^*)$.

Since $P(\boldsymbol{\gamma}|\boldsymbol{r},\boldsymbol{\alpha}^*) = \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{\gamma}}^*, \Sigma_{\boldsymbol{\gamma}}^*)$, we have [16],

$$h(\boldsymbol{\gamma}|\boldsymbol{r},\boldsymbol{\alpha}^*) = \frac{1}{2}\log\left(|2\pi e\Sigma_{\boldsymbol{\gamma}}^*|^+\right) \tag{19}$$

$$= \frac{1}{2}\log\left(|\boldsymbol{B}^T\boldsymbol{B}||2\pi e\Sigma_{\boldsymbol{w}}^*|\right) \tag{20}$$

$$= \frac{1}{2}\log\left(|\boldsymbol{B}^T\boldsymbol{B}|\right) + h(\boldsymbol{w}|\boldsymbol{r},\boldsymbol{\alpha}^*) \tag{21}$$

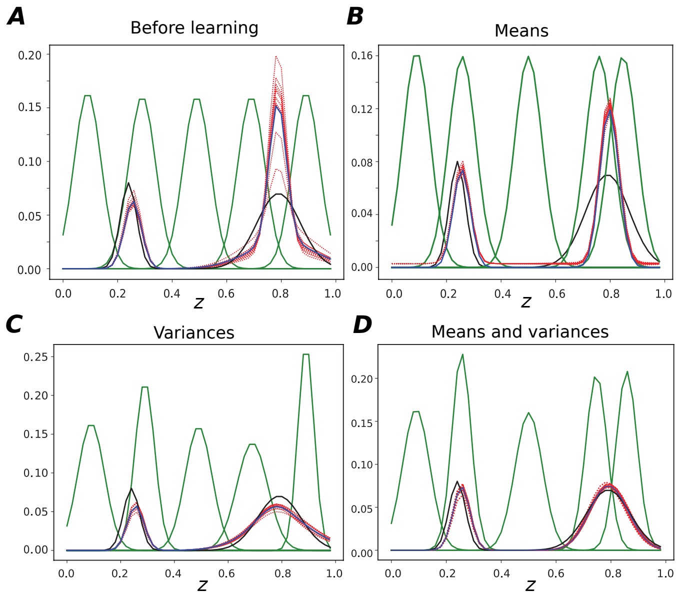

Fig. 2: The effect of learning the means and/or variances of Gaussian encoding functions on the decoding precision and entropy of the distribution function. A) the encoding functions (green) before learning. The true distribution is shown in black. The conditional mean of the posterior distribution, $\boldsymbol{\mu}^*_{\boldsymbol{\gamma}}$, is depicted in blue. Some generated samples of the posterior are shown in red. B) Learning the means of the Gaussian encoding functions. C) Learning only the variances of the encoding functions. D) Learning both the means and variances of the encoding functions. In all the simulations, $\alpha_0 = 100$.

where $|.|^+$ is the pseudo-determinant. Also,

$$h(\boldsymbol{w}|\boldsymbol{r},\boldsymbol{\alpha^*}) = \frac{1}{2}\log\left(|2\pi e \Sigma^*_{\boldsymbol{w}}|\right) \tag{22}$$

$$= -\frac{1}{2}\log\left(|(2\pi e)^{-1}(\alpha_0 \boldsymbol{B}^T\boldsymbol{\Phi}^T\boldsymbol{\Phi}\boldsymbol{B} + D_{\boldsymbol{\alpha^*}})|\right) \tag{23}$$

In (21), the first term is constant, therefore, to minimize the entropy $h(\boldsymbol{\gamma}|\boldsymbol{r},\boldsymbol{\alpha^*})$, we need to minimize $h(\boldsymbol{w}|\boldsymbol{r},\boldsymbol{\alpha^*})$. Hence, the parameters of the encoding functions should be adjusted such that the matrix of encoding functions, $\boldsymbol{\Phi}$, minimizes (23).

Let $U = \alpha_0 \boldsymbol{B}^T\boldsymbol{\Phi}^T\boldsymbol{\Phi}\boldsymbol{B} + D_{\boldsymbol{\alpha^*}}$, and c be an arbitrary parameter of the encoding functions,

$$\frac{\partial h(\boldsymbol{w}|\boldsymbol{r},\boldsymbol{\alpha^*})}{\partial c} = -\frac{1}{2\ln 2}\mathrm{Tr}(U^{-1}\frac{\partial U}{\partial c}), \tag{24}$$

where Tr(.) is the trace of the matrix. Moreover,

$$\frac{\partial U}{\partial c} = \alpha_0 \boldsymbol{B}^T(\frac{\partial \boldsymbol{\Phi}^T}{\partial c}\boldsymbol{\Phi} + \boldsymbol{\Phi}^T\frac{\partial \boldsymbol{\Phi}}{\partial c})\boldsymbol{B}. \tag{25}$$

Therefore,

$$\frac{\partial h(\boldsymbol{w}|\boldsymbol{r},\boldsymbol{\alpha^*})}{\partial c} = -\frac{1}{2\ln 2}\mathrm{Tr}\Big((\alpha_0 \boldsymbol{B}^T\boldsymbol{\Phi}^T\boldsymbol{\Phi}\boldsymbol{B} + D_{\boldsymbol{\alpha^*}})^{-1} \times \alpha_0 \boldsymbol{B}^T(\frac{\partial \boldsymbol{\Phi}^T}{\partial c}\boldsymbol{\Phi} + \boldsymbol{\Phi}^T\frac{\partial \boldsymbol{\Phi}}{\partial c})\boldsymbol{B}\Big). \tag{26}$$

IV. Learning in 1D latent spaces

We assume that the latent variable Z corresponds to a one-dimensional space, and the latent state $Z = j$, $j \in \{1, 2, \ldots, S\}$, is mapped to the value $z_j = \frac{j-1}{S} + \frac{1}{2S}$ in the interval $[0, 1]$. For example, Z can denote the discrete location of an animal on a linear track. At each location, the animal receives new sensory information and forms a distributional belief about its location, which is represented by the DDC values. The set of encoding functions and the DDC values determine the amount of uncertainty about the belief. By modifying the encoding functions, the brain can reduce this uncertainty.

A. Gaussian encoding functions

First we assume that the encoding functions are Gaussian functions, with initial mean μ_i and the standard deviation σ_i,

$$\boldsymbol{\Phi}(i,j) = \frac{1}{\sigma_i\sqrt{2\pi}}e^{\frac{-(z_j-\mu_i)^2}{2\sigma_i^2}} \tag{27}$$

where $i \in \{1, 2, \ldots, K\}$ and $j \in \{1, 2, \ldots, S\}$. We can learn the parameters of the encoding functions, means and variances, using (26). The derivative of $\boldsymbol{\Phi}$ with respect to μ_k is derived from

$$\frac{\partial \boldsymbol{\Phi}(i,j)}{\partial \mu_k} = \begin{cases} \frac{(z_j-\mu_i)}{\sigma_i^2}\boldsymbol{\Phi}(i,j) & \text{if} \quad i = k \\ 0 & \text{if} \quad i \neq k \end{cases} \tag{28}$$

and we use the gradient descent to find the (local) optimum of the mean of each encoding function. The mean of the k'th encoding function is updated by

$$\mu_k(n) = \mu_k(n-1) - \lambda_\mu(n)\frac{\partial h(\boldsymbol{w}|\boldsymbol{r},\boldsymbol{\alpha^*})}{\partial \mu_k} \tag{29}$$

where λ_μ is the learning rate, and $\frac{\partial h(\boldsymbol{w}|\boldsymbol{r},\boldsymbol{\alpha^*})}{\partial \mu_k}$ is calculated from (26) and (28). After each update, $\boldsymbol{\Phi}$ is modified according to the new encoding functions, and the vector of DDC values, $\boldsymbol{r}$, is altered by the new $\boldsymbol{\Phi}$, though the belief $\boldsymbol{\gamma}$ may remain the same. We then derive the optimal precision values $\boldsymbol{\alpha}^*$ for the new $\boldsymbol{r}$, and repeat the steps until convergence. To avoid oscillations of the optimization algorithm around the optimal point, we use an exponential decay for the learning rate. The learning rate at time n, is derived from $\lambda_\mu(n) = \lambda_\mu(0)e^{-\kappa n}$, where κ is the decay constant.

We simulated the learning process for 5 Gaussian encoding functions (Fig. 2). We assume that the true distribution function is a mixture of Gaussians (black line). In Fig. 2A, we show the DDC encoding functions before learning (green lines). As described in Section III, we derive the posterior distribution over the distributional belief. The mean of the distributional belief $\boldsymbol{\gamma}$ given the DDC values, $\boldsymbol{\mu}^*_{\boldsymbol{\gamma}}$ (blue line), can not capture the true distribution function. The generated samples of $\boldsymbol{\gamma}$ (red lines) demonstrate high uncertainty around $z = 0.8$. After learning the means of the encoding functions (Fig. 2B), the amount of uncertainty reduces significantly, and the distribution function is reconstructed more accurately.

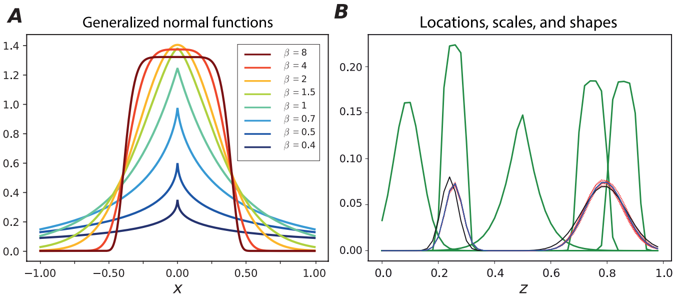

Fig. 3: Learning the shape of encoding functions. A) Generalized normal functions. The shape parameter, β, modifies the shape of the function. B) Learning the locations, scales, and shapes of the encoding functions. The line colors are as in Fig. 2.

We can also use the gradient descent to find the standard deviation of the Gaussian encoding functions,

$$\frac{\partial \mathbf{\Phi}(i,j)}{\partial \sigma_k} = \begin{cases} \frac{(z_j-\mu_i)^2-\sigma_i^2}{\sigma_i^3}\mathbf{\Phi}(i,j) & \text{if} \quad i = k \\ 0 & \text{if} \quad i \neq k \end{cases} \tag{30}$$

and the standard deviation is updated by

$$\sigma_k(n) = \sigma_k(n-1) - \lambda_\sigma(n)\frac{\partial h(\boldsymbol{w}|\boldsymbol{r},\boldsymbol{\alpha}^*)}{\partial \sigma_k}. \tag{31}$$

Learning the variances of the encoding functions results in lower uncertainty and a more accurate reconstruction of the true distribution (Fig. 2C). The amount of uncertainty is minimized by learning both the means and variances of the encoding functions (Fig. 2D), and the calculated mean of the posterior, $\boldsymbol{\mu}_{\boldsymbol{\gamma}}^*$, fits the true distribution with high precision.

B. Generalized normal functions

We can also learn the shape of the encoding functions. We assume that the set of encoding functions consists of Generalized normal functions, with the parameters, μ (location), α (scale), and β (shape),

$$f(x) = \frac{\beta}{2\alpha\Gamma(1/\beta)}e^{-(|x-\mu|/\alpha)^\beta}. \tag{32}$$

In Fig. 3, generalized normal functions have been plotted for different shape parameters. We calculate the derivative of the entropy with respect to the shape parameter of the encoding function. The entry (i,j) of the matrix of the encoding functions is

$$\mathbf{\Phi}(i,j) = \frac{\beta_i}{2\alpha_i\Gamma(1/\beta_i)}e^{-(|z_j-\mu_i|/\alpha_i)^{\beta_i}} \tag{33}$$

where $i \in \{1,2,\ldots,K\}$ and $j \in \{1,2,\ldots,S\}$. The derivative of $\mathbf{\Phi}$ with respect to β_i is derived from

$$\frac{\partial \mathbf{\Phi}(i,j)}{\partial \beta_i} = \frac{1}{2\alpha_i}\left(\frac{1}{\Gamma(\frac{1}{\beta_i})} - \frac{\beta_i\frac{d}{d\beta_i}\Gamma(\frac{1}{\beta_i})}{\Gamma^2(\frac{1}{\beta_i})}\right)e^{-(|z_j-\mu_i|/\alpha_i)^{\beta_i}} \\ - \frac{\beta_i}{2\alpha_i\Gamma(\frac{1}{\beta_i})}e^{-(|z_j-\mu_i|/\alpha_i)^{\beta_i}}\frac{\partial}{\partial \beta_i}\left((|z_j-\mu_i|/\alpha_i)^{\beta_i}\right) \tag{34}$$

The derivative of the logarithm of $\Gamma(x)$ is the digamma function $\digamma(x)$,

$$\digamma(x) = \frac{d}{dx}\ln\left(\Gamma(x)\right) = \frac{\Gamma'(x)}{\Gamma(x)}. \tag{35}$$

Therefore, from (34) and (35),

$$\frac{\partial \mathbf{\Phi}(i,j)}{\partial \beta_i} = \mathbf{\Phi}(i,j)\Bigg(\frac{\beta_i + \digamma(\frac{1}{\beta_i})}{\beta_i^2} - (|z_j-\mu_i|/\alpha_i)^{\beta_i} \\ \times \ln(|z_j-\mu_i|/\alpha_i)\Bigg) \tag{36}$$

The derivative with respect to the scale parameter is derived from

$$\frac{\partial \mathbf{\Phi}(i,j)}{\partial \alpha_i} = \mathbf{\Phi}(i,j)\left(\frac{-1}{\alpha_i} + \frac{\beta_i}{\alpha_i}(|z_j-\mu_i|/\alpha_i)^{\beta_i}\right), \tag{37}$$

and with respect to the location parameter,

$$\frac{\partial \mathbf{\Phi}(i,j)}{\partial \mu_i} = \mathbf{\Phi}(i,j)\frac{\beta_i}{\alpha_i}(|z_j-\mu_i|/\alpha_i)^{\beta_i-1}\text{sgn}(z_j-\mu_i), \tag{38}$$

where $\text{sgn}(x)$ is the sign function.

We can therefore learn the location, scale and shape of the generalized normal functions. It is shown in Fig. 3B that learning the shapes of the DDC encoding functions reduces the uncertainty about the distribution function and produces a precise reconstruction of the true distribution. These results propose an explanation for different shapes of neuronal tuning functions: distinct tuning shapes have been evolved to reduce the uncertainty of the brain about the distribution over the variables of interest.

C. Dynamics of encoding functions

Sensory signals change constantly, and the inferred distribution function over the latent variable is time-variant. The learning mechanism of the DDC encoding functions should take into account the time-varying characteristics of the distribution function. We extend our model to dynamic learning and show how the encoding functions are gradually modified to reduce the instantaneous uncertainty about the current distributional belief. This analysis reveals the dynamical properties of encoding functions and is employed to study the dynamics of neuronal tuning functions.

We explain the dynamics of encoding functions through examples of spatial navigation. Let's assume that an animal is moving on a linear track. At each location on the track, the navigational uncertainty of the animal is characterized by the posterior distribution over location, and is represented by a DDC code. The DDC encoding functions are initially located randomly (Fig. 4A). When the animal is moving on the track, we assume that the posterior distribution over the animal's location sweeps the latent space uniformly (Fig. 4B). At each location, the posterior distribution is represented by the corresponding vector of DDC values, $\boldsymbol{r}$; the optimal precision vector $\boldsymbol{\alpha}^*$ is calculated, and the encoding functions are slightly updated to reduce the instantaneous uncertainty $h(\boldsymbol{\gamma}|\boldsymbol{r},\boldsymbol{\alpha}^*)$. Then the animal moves to a different location,

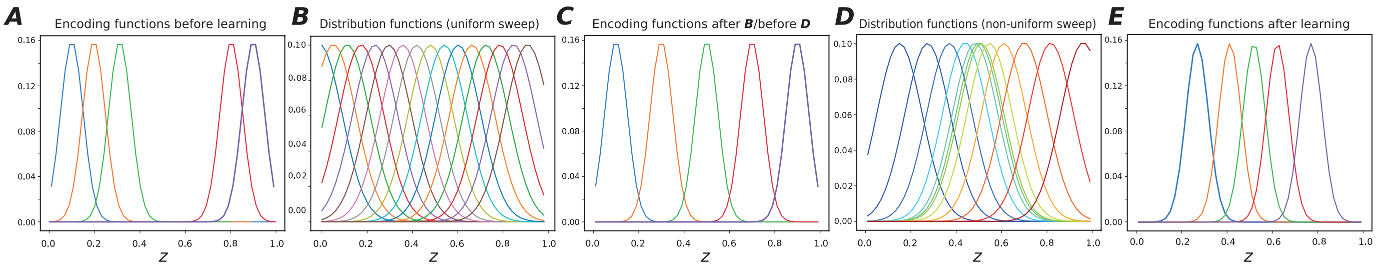

Fig. 4: Learning the encoding functions for dynamic distribution functions. A) Initial location of encoding functions. B) The distribution function sweeps the latent space uniformly. The mean of the distribution function linearly increases by time from $z = 0$ to $z = 1$, and resets to zero after reaching $Z = 1$. C) The encoding functions after learning the sequence of distributions in B. D) Distribution function sweeps the latent space non-uniformly, with more distributions around the center. The mapping from time to the mean of the distribution function is non-linear, and its slope increases by the absolute distance from $z = 0.5$. E) Encoding functions after learning the sequence of distributions in D.

the posterior over location is updated, a new vector of DDC values is formed, and the procedure continues. After hundreds of laps, this learning algorithm leads to a uniform placement of the DDC encoding functions (Fig. 4C). This result has been already observed in the hippocampal place cells (neurons in CA1 and CA3 regions of the hippocampus that are activated when the animal is in a specific location). It has been shown that at the initial phase of exploration, the place cells are randomly located. However, after enough trials, the place cells tile the environment uniformly [17].

When there is a reward in the environment, the animal spends more time around the estimated reward location (non-uniform sweep). The sequence of posterior distributions over location has therefore more entries around the reward zone (Fig. 4D). This non-uniform sweep will create a higher density of encoding functions around the concentration point (Fig. 4E). The encoding functions are updated to reduce the uncertainty about distribution functions, and since distribution functions are around the reward location most of the time, the learning algorithm pushes the encoding functions towards the reward zone. This phenomenon has been also validated in the experiments: when the animal spends more time near the goal location, the place cells shift toward the goal [18].

V. Discussion

Perception appears to be built on internalised probabilistic beliefs about features of the external world. One hypothesis about how such beliefs may be represented in the brain is the distributed distributional code (DDC), in which neural activity corresponds to expectations of a set of encoding functions under the internal belief. Here we asked how these encoding functions might be adapted to increase the fidelity of the representation. We assumed that the beliefs to be encoded are sparse in a known basis, and modeled the DDC values using hierarchical priors. We derived the conditional uncertainty about the distributional belief given the DDC values, and then minimised this uncertainty with respect to parameters in the encoding functions. This general learning rule can be exploited in various applications. We showed that the means, variances, and shapes of the encoding functions obtained in this way suggest links to known neuronal tuning properties.

Our framework can be easily extended to higher dimensions. If the latent variable is n-dimensional, we parametrize a set of n-dimensional encoding functions and learn each parameter of the encoding functions through (26). Moreover, the learning algorithm that was derived here is applicable to the optimisation of classic tuning functions in sensory areas and the other encoding functions that represent a sparse function linearly.

The set of basis functions, $\boldsymbol{B}$, that is used in this paper consists of normal functions with different widths and means; the basis matrix $\boldsymbol{B}$, however, can be replaced by other functions, such as discrete cosine transform basis functions.

The derived sparse coefficients, $\boldsymbol{w}$, do not necessarily correspond to a normalized non-negative distributional belief $\boldsymbol{\gamma}$. However, neither optimisation of the encoding functions, nor downstream computation, depends on explicit decoding of the distribution function. The minimization of the entropy provides direct learning rules for the parameters of the encoding functions, while many computations with DDC representations depend on simple linear combinations.

Probabilistic inference in the brain is not limited to sensory signals. The brain must deal with uncertainty during action, cognition, and perception. Action planning and motor execution, decision making, and a wide variety of cognitive tasks are fundamentally probabilistic and may depend on distributional representations such as the DDC. We can therefore employ the results of this paper to derive efficient DDC encoding functions that could underlie different cognitive tasks across various regions of the brain. Thus, this approach may help us to gain a general understanding of the dynamics of neuronal tuning functions from the perspective of DDC Bayesian computation.

Acknowledgement

We would like to thank the reviewers for their helpful comments that have improved the quality of the paper. This work was supported by the Gatsby Charitable Foundation and the Simons Foundation (SCGB 543039).

References

[1] D. C. Knill and A. Pouget, "The bayesian brain: the role of uncertainty in neural coding and computation," *Trends in Neurosciences*, vol. 27, no. 12, pp. 712–719, 2004.

[2] W. J. Ma, J. M. Beck, P. E. Latham, and A. Pouget, "Bayesian inference with probabilistic population codes," *Nature neuroscience*, vol. 9, no. 11, pp. 1432–1438, 2006.

[3] J. Beck, W. Ma, P. Latham, and A. Pouget, "Probabilistic population codes and the exponential family of distributions," *Progress in brain research*, vol. 165, pp. 509–519, 2007.

[4] P. O. Hoyer and A. Hyvärinen, "Interpreting neural response variability as monte carlo sampling of the posterior," pp. 293–300, 2003.

[5] G. Orbán, P. Berkes, J. Fiser, and M. Lengyel, "Neural variability and sampling-based probabilistic representations in the visual cortex," *Neuron*, vol. 92, no. 2, pp. 530–543, 2016.

[6] R. S. Zemel, P. Dayan, and A. Pouget, "Probabilistic interpretation of population codes," *Neural computation*, vol. 10, no. 2, pp. 403–430, 1998.

[7] M. Sahani and P. Dayan, "Doubly distributional population codes: simultaneous representation of uncertainty and multiplicity," *Neural Computation*, vol. 15, no. 10, pp. 2255–2279, 2003.

[8] E. Vértes and M. Sahani, "Flexible and accurate inference and learning for deep generative models," *Advances in Neural Information Processing Systems*, 2018.

[9] ——, "A neurally plausible model learns successor representations in partially observable environments," *Advances in Neural Information Processing Systems*, vol. 32, pp. 13 714–13 724, 2019.

[10] L. K. Wenliang and M. Sahani, "A neurally plausible model for online recognition and postdiction in a dynamical environment," *Advances in Neural Information Processing Systems*, p. 672089, 2020.

[11] M. E. Tipping, "The relevance vector machine," pp. 652–658, 2000.

[12] ——, "Sparse bayesian learning and the relevance vector machine," *Journal of machine learning research*, vol. 1, no. Jun, pp. 211–244, 2001.

[13] S. Ji, Y. Xue, and L. Carin, "Bayesian compressive sensing," *IEEE Transactions on signal processing*, vol. 56, no. 6, pp. 2346–2356, 2008.

[14] S. D. Babacan, R. Molina, and A. K. Katsaggelos, "Bayesian compressive sensing using laplace priors," *IEEE Transactions on image processing*, vol. 19, no. 1, pp. 53–63, 2009.

[15] D. J. MacKay, "Bayesian interpolation," *Neural computation*, vol. 4, no. 3, pp. 415–447, 1992.

[16] T. M. Cover and J. A. Thomas, "Elements of information theory second edition solutions to problems," *Internet Access*, pp. 19–20, 2006.

[17] S. Kim, D. Jung, and S. Royer, "Place cell maps slowly develop via competitive learning and conjunctive coding in the dentate gyrus," *Nature communications*, vol. 11, no. 1, pp. 1–15, 2020.

[18] S. A. Hollup, S. Molden, J. G. Donnett, M.-B. Moser, and E. I. Moser, "Accumulation of hippocampal place fields at the goal location in an annular watermaze task," *Journal of Neuroscience*, vol. 21, no. 5, pp. 1635–1644, 2001.