

# Processing and transmission of confidence in recurrent neural hierarchies

Alexander Geppert

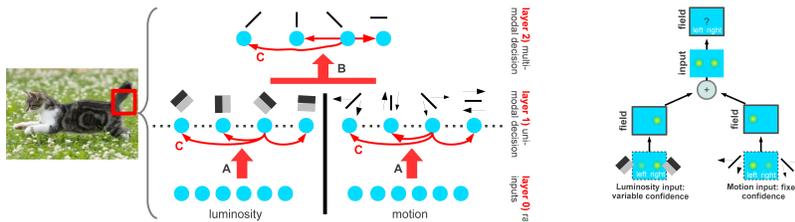
Received: date / Accepted: date

**Abstract** This article addresses the construction of hierarchies from dynamic attractor networks. We claim that such networks, e.g., dynamic neural fields (DNFs), contain a *data model* which is encoded in their lateral connections, and which describes typical properties of afferent inputs. This allows to infer the most likely interpretation of inputs, robustly expressed through the position of the attractor state. The principal problem resides in the fact that positions of attractor states alone do not reflect the quality of match between input and data model, termed *decision confidence*. In hierarchies, this inevitably leads to final decisions which are not Bayes-optimal when inputs exhibit different degrees of ambiguity or conflict, since the resulting differences in confidence will be ignored by downstream layers. We demonstrate a solution to this problem by showing that a correctly parametrized DNF layer can *encode* decision confidence into the latency of the attractor state in a well-defined way. Conversely, we show that input stimuli gain competitive advantages w.r.t. each other as a function of their relative latency, thus allowing downstream layers to *decode* attractor latency in an equally well-defined way. Putting these encoding and decoding mechanisms together, we construct a 3-stage hierarchy of DNF layers and show that the top-level layer can take Bayes-optimal decisions when the decisions in the lowest hierarchy levels have variable degrees of confidence. In the discussion, we generalize these findings, suggesting a novel possibility to represent and manipulate probabilistic information in recurrent networks without any need for log-encoding, just using the biologically well-founded effect of response latency as an additional coding dimension.

**Keywords** Recurrent neural networks · Neural coding · Bayesian inference

---

A.Geppert  
ENSTA ParisTech  
Boulevard des Maréchaux, 91762 Palaiseau, France  
Tel.: +33 1818 72041  
E-mail: alexander.geppert@ensta-paristech.fr



**Fig. 1** Motivation and basic setting for the presented work. **Left:** Perceptual decision hierarchy making use of response latency for optimal decision making. We imagine a hierarchical cortical network analyzing a small image patch (red rectangle) and analyzing it along two (or several) visual modalities as it has been demonstrated in lower visual areas of mammals. The (fixed) feed-forward connections between layers (**A,B**) cause feature selectivity in layer 1/2 neurons (indicated by small symbols for some neurons) whereas the (fixed) lateral connections (**C**) contain the data model for inputs to layers 1/2. Violations of this data model, e.g., by strong co-activation of unimodal layer 1 neurons, will amplify unimodal response latency, thus reducing influence on the multimodal integration layer 2. **Right:** actual neural hierarchy considered in this article. Layers (implemented by dynamic neural fields) and visual modalities correspond to the left diagram but stimulus structure has been simplified to admit only two possible stimulus types in each modality. For simplicity, these will often be termed the "left" and "right" stimulus. During experiments, one modality will always receive the same layer 0 inputs, whereas the other modality will receive inputs leading to variable decision confidence, and the resulting decisions in layers 1 and 2 will be observed.

## 1 Introduction

The issue of constructing deep neural hierarchies has recently received considerable interest, sparked mainly by research in deep belief networks (DBNs, [17,2]) and convolutional hierarchies[22]. These approaches are successful in several application scenarios but do not make use of the fact that each layer in biological processing hierarchies has strong lateral connections, leading to complex non-linear dynamics within a single layer. Therefore, one may speculate that the computational potential of processing hierarchies constructed from recurrent neural layers might be even higher than that of present-day DBNs. Building such recurrent hierarchies is however tricky due to the inherent non-linear behavior of recurrent layers, and a way needs to be found to make the nonlinear dynamics work 'for us' instead of 'against us'.

In order to illustrate the basic issues when constructing deep hierarchies using recurrent neural layers, we will use perceptual decision making as an example, considering a decision-making task depicted in Fig. 1. It exhibits a two-level hierarchy where the high-level decision integrates two lower-level decisions about the interpretation of complementary sensory input flows. Such interpretations are always based on an implicit model of the "true" nature of inputs which we denote a *data model*. Such data models are highly important for real-world processing as inputs may be corrupted by (structured and complex) noise. Using data models, however, the most likely interpretation given the data model can be computed before transmission to subsequent hierarchy levels, thus removing noise and increasing signal quality. Such in-

terpretations amount to *decision making* about which parts of a stimulus, if any, to discard and which ones to keep. Consequently, each perceptual decision can be attributed a *confidence value* depending on the data model. For stimuli which have a common cause in the external world, as in the example of Fig. 1, only a single of the feature-selective patches can be active at the same time, that is to say: the data model must assign a lower confidence to inputs with, e.g., ambiguity or conflict. As inputs to the lower hierarchy levels may cause different degrees of decision confidence, it is imperative to take these confidences into account if the high level is to take Bayes-optimal decisions!

However, as recurrent attractor networks usually converge to an attractor state representing only the strongest input, no information about the confidence of lower-level decisions remains in the converged network states. To remedy this, we investigate a possibility of including confidence information in neural responses, in such a way that confidence can easily be encoded, and equally easily be decoded by subsequent layers. As it turns out, it is the non-linear dynamics themselves that can, when parametrized correctly, naturally implement such a mechanism. Due to the nonlinear build-up of membrane potential in model neurons, as well as due to competitive interactions, input confidence according to Fig. 2 can be translated into response latency, i.e., the time from stimulus onset to the development of significant activity. Vice versa, latency differences of inputs cause differences in corresponding membrane potentials, which give different neural populations different influences in the competitive selection process.

Summarizing this, we propose that recurrent neural connections define a data model for interpreting input stimuli, and that the confidence of decisions under this data model can be encoded into neural response latency. Response latency can thus be considered a secondary coding dimension in addition to, e.g., firing rate<sup>1</sup>, encoding and transporting a confidence measure across hierarchy levels. This mechanism effectively multiplies the information carried by each neural layer without requiring additional resources, and can be extremely useful in real-world scenarios where confidence measurements are important due to incomplete, noisy and contradictory sensory inputs.

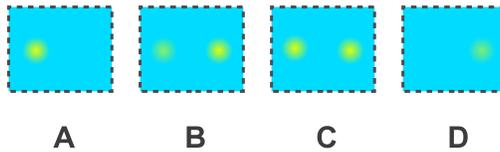
### 1.1 Biological background on response latency and confidence

The effect of response latency is ubiquitous in biology, and there is converging evidence from both physiological[27, 30, 19, 25] and behavioral [16, 5] investigations that it plays a role in the neural encoding of information.

Response latency is linked to different causes, some of which are overlapping: neurons in the striate cortex, for example, encode stimulus contrast into response latency[30]. On the behavioral side, it has been found that decision making processes typically take longer depending on the number of conflicting

---

<sup>1</sup> This article uses a rate-coded model for simplicity, but we do not wish to exclude spiking models, where the effect of response latency has been documented as well[33, 36]



**Fig. 2** Simplified input stimuli leading to different degrees of decision confidence. A) maximal confidence: a single stimulus of maximal amplitude B) reduced confidence due to ambiguity/conflict: a second stimulus is present but of weaker peak amplitude, thus still allowing a decision C) low confidence due to strong ambiguity/conflict: two stimuli of equal peak amplitude do not allow a decision at all D) lack of evidence: a single stimulus of sub-maximal peak amplitude. The smaller the peak amplitude, the lesser the confidence.

alternatives[16], conceivably reflecting increased response latency on the neural level. Similar effects have been observed in language processing[5], where the ambiguity, i.e., the number of different interpretations, gives rise to delayed responses. The terms 'contrast', 'conflict' and 'ambiguity' that are used in the literature for the causes of response latency denote very similar concepts. In this article, we will characterize stimuli by the terms "evidence" (or lack thereof) as well as "conflict/ambiguity". Examples for inputs exhibiting these properties are shown in Fig. 2.

Since biological neural networks are strongly hierarchical, the existence of response latency automatically implies the existence of input latency at higher hierarchy levels. In [19], it is speculated how input latency could be decoded in downstream neural populations; in this contribution we propose just such a mechanism which is computationally simple and biologically plausible.

## 1.2 Related modeling work

As a network model, we choose the dynamic neural field (DNF) model[1,34], which is a recurrent, rate-coded model originally conceived to describe cortical processing. Today, it is widely used for modeling memory [18,39], decision making [7,11], human-robot interaction[3,32]. The model exhibits many attractive properties, especially continuous attractor dynamics which is desirable for, e.g., generating smooth robot control commands, or for modeling the dynamics of decision making[7].

On the modeling side, response latency is the key idea behind the rank order coding model[13,21] which posits that the precise timing of arriving spikes, relative to stimulus onset, carries information about the relative importance of the represented concepts. Some models[36] even consider the first arriving spikes at the exclusion of all others. Our findings are compatible with all of these models, for although our investigation is based on rate-coded model neurons, the key findings that certain inputs lead to quicker responses, and that quicker responses dominate downstream processing (as demonstrated for spiking networks in [33]), are at the core of this investigation.

When considering the larger implications of how biological neurons might compute and manipulate probabilistic information, there exists a large body of literature[23,38,15,20,38,8,29]. There seems to be agreement that neural populations represent more than just values; indeed, most authors explicitly assume that neural population activity is related, in various proposed fashions, to probability or "belief" distributions[23,38,20,8,29]. A very influential idea posits that neural activity is related to log-probability[8,29,15], which is attractive because the multiplication of populations activities (which is considered necessary for integration and Bayesian inference) then amounts to a simple summation which neurons can do easily. However, other authors have questioned the practicability of this scheme[23] as well as the basic assumption that Bayesian inference is indeed implemented by multiplying the *population activities* in a neuron-by-neuron fashion. To the first point, it is claimed in [23] that hierarchical inference steps, i.e., using the result of one neural layer as the basis for another one, require a re-encoding at each level which seems unfeasible. To the second point, it is questioned[23] that a sensory measurement indeed constitutes a probability distribution in the usual sense. Instead the authors claim that each neuron's firing rate represents the realization of a random variable governed by a Poisson-like probability distribution determined by the match of afferent input with that neuron's preferred stimulus. Thus, the mean of each neuron's activity is deterministic and governed by the degree of match. According to [23], the fact that a neuron represents the realization of a random variable, and not simply a discrete bin in a probability distribution, makes it questionable that multiplying neural activities is a statistically sensible thing to do in any case.

On the other hand, there are models of population encoding and Bayesian inference which do not use log-encoding[23,10,28,38] distributions, which have to resort to more complicated schemes like attractor networks[10] or assumptions about noise distributions[23].

The presented work evidently does not use log-encoding for representing and processing probabilistic information; we will discuss its similarities and differences to related research in Sec. 4.

### 1.3 Research questions and article outline

The main point of this article is the construction of processing hierarchies that take into account confidence for optimal decision making, see Fig. 1. Working with very simple input stimuli to demonstrate essential mechanisms, this article poses and answers three research questions:

**Q1: Can recurrent dynamics encode decision confidence into response latency?** Here, we ask whether decision confidence, i.e., the degree of match between an input stimulus to the data model encoded by the recurrent connections, can be unambiguously translated into response latency. For successful encoding, we demand that the response latency should be a

monotonous function of decision confidence. In other words: the less confident the input, the greater the response latency.

**Q2: Can recurrent dynamics decode response latency?** For decoding, we demand that the ability of arriving inputs to influence attractor formation is a monotonously decreasing function of their latency. Put briefly: later-coming inputs should be less likely to influence or win the competition process.

**Q3: Can encoding and decoding steps be successfully coupled?** Here, we are interested in the fact whether two neural fields, one encoding and one decoding decision confidence expressed as response latency, can be coupled such that the resulting decision takes into account confidence differences correctly

These questions will be addressed by simulating the neural hierarchy depicted in Fig. 1 using the DNF model [1, 34]. After a review of the employed DNF model and its numerical simulation<sup>2</sup> in Sec. 2, we will more precisely define the artificial input stimuli in Sec. 2.3 as well as the assumed data model and its analytical formulation in Sec. 2.4. The experiments conducted in Sec. 3 correspond directly to the research questions raised here, and in Sec. 4 it will be discussed to what extent the experimental results give answers to these questions. In Sec. 4, it will also be attempted to generalize the obtained results, suggesting how recurrent networks can encode, and possibly learn, internal data models. A critical discussion of the impact of this work on neural coding, as well as a discussion of limitations and possible future work concludes the article.

## 2 Methods

We base our investigation on the dynamic neural field model [1] which was originally proposed to describe pattern formation in the visual cortex. Essentially, dynamic neural fields are a class of recurrent neural network models that have been extensively used for modeling cognitive phenomena like decision making [6], motor planning [11], spatial cognition [18], eye movement preparation [37, 31] and object recognition [12, 9]. Basic elements are simple dynamic-state neurons, a fixed lateral connectivity, and a (usually sigmoid) non-linearity.

### 2.1 Model equations

We use a slightly more general version of the original model in the sense that afferent and lateral terms can be weighted differently using the coefficients

---

<sup>2</sup> Python/C code implementing all simulations of this article is available under [www.geppert.net/alexander](http://www.geppert.net/alexander)

$\alpha, \beta$ :

$$\begin{aligned} \tau \dot{u}(\mathbf{x}, t) = & -u(\mathbf{x}, t) + \alpha S(\mathbf{x}, t) \\ & + \beta \int w(\mathbf{x} - \mathbf{x}') f[u(\mathbf{x}', t)] d\mathbf{x}' + \gamma \sigma(\mathbf{x}, t) + h \end{aligned} \quad (1)$$

$$\text{where } f[u(\mathbf{x}, t)] = \frac{1}{1 + \exp\left(\frac{-2(u(\mathbf{x}, t) - \theta)}{\nu}\right)}$$

Here, the quantity  $u(\mathbf{x}, t)$  represents the membrane potential of the field at time  $t$  and position  $\mathbf{x}$ ,  $S(\mathbf{x}, t)$  the afferent input,  $w(\mathbf{x} - \mathbf{x}')$  the fixed lateral *interaction kernel*,  $f[u]$  the non-linearity or *transfer function*, and  $\sigma(\mathbf{x}, t)$  normally distributed white noise.  $\tau$  determines the time scale of field evolution, and  $h$  is the *resting potential*, i.e., the equilibrium potential in case of no input. We choose a sigmoid transfer function, parametrized by a threshold and a gain value:  $\theta, \nu$ . The coefficients  $\alpha, \beta$  and  $\gamma$  respectively determine the contribution of the afferent input, the lateral recurrent interactions and the noise. The interaction kernel  $w(\mathbf{x} - \mathbf{x}')$  is usually chosen to be symmetric:  $w(\mathbf{x} - \mathbf{x}') = a_0 G_{\mu=0, \sigma_{\text{on}}}(\mathbf{x} - \mathbf{x}') - b_0 G_{\mu=0, \sigma_{\text{off}}}(\mathbf{x} - \mathbf{x}') - c_0$ , where  $G_{\mu=0, \sigma}(\mathbf{x})$  denotes a Gaussian with mean  $\mu$  and standard deviation  $\sigma$ , and  $\sigma_{\text{on}} < \sigma_{\text{off}}$ . The constants  $a_0, b_0, c_0$  are chosen suitably to achieve the desired level of local excitation/inhibition ( $a_0, b_0$ ) as well as global inhibition ( $c_0$ ). To ensure numerical stability, we clip the neural field potentials  $u(\mathbf{x}, t)$  whenever they exceed the range defined by  $[u_{\min}, u_{\max}]$ .

## 2.2 Numerical simulation

For performing numerical computations, neural fields potentials are discretized to a grid of  $N \times N$  "neurons", denoted by  $\hat{u}(\mathbf{x}, t)$ . The lateral interaction filter  $\hat{w}$  is discretized as well, having a width of  $5\sigma_{\text{off}}$  elements, while global inhibition is obtained by summing all elements in the discretized field. Thus, for discretized neural fields, the update equation reads

$$\begin{aligned} \hat{u}(\mathbf{x}, t + 1) = & \left(1 - \frac{1}{\tau}\right) \hat{u}(\mathbf{x}, t) + \frac{1}{\tau} \times \\ & \times \left[ \alpha \hat{S}(\mathbf{x}, t) + \beta \sum_{|x'|, |y'| < 2.5\sigma_{\text{off}}} \hat{w}(x', y') f \left[ \hat{u} \left( \begin{pmatrix} x - x' \\ y - y' \end{pmatrix}, t \right) \right] + \right. \\ & \left. + \beta c_0 \sum_{\mathbf{x}'} f[u(\mathbf{x}', t)] + \gamma \hat{\sigma}(\mathbf{x}, t) + h \right] \end{aligned} \quad (2)$$

## 2.3 Used stimuli

As mentioned in Fig. 1, we will use synthetic input stimuli  $\mathbf{S} \equiv S(\mathbf{x}, t)$  in our experiments in order not to complicate the demonstration of the desired effects

by subtleties of real-world data processing. Stimuli  $\mathbf{S}$  consist of two Gaussian "bubbles", at two fixed positions, with equal variances (see, e.g., [34]), which have peak values  $A_1, A_2 \in [0, 1]$ . We furthermore assume that  $\mathbf{S}$  is a version of the "true" underlying stimulus  $\mathbf{M}$  which is corrupted by structured noise, and that  $\mathbf{M}$  contains only a single Gaussian at one of the two allowed locations.  $\mathbf{M}$  can therefore be described by two numbers  $M_1, M_2 \in \{0, 1\}$ ,  $M_1 + M_2 = 1$ , indicating the amplitude of Gaussians at each of the two locations. Example stimuli  $\mathbf{S}$  are shown in Fig. 2. For simplicity, we will often refer to a stimulus where only one Gaussian is present as "left" or "right" depending on the Gaussian's location.

## 2.4 Implicit and explicit data model formulation

By the assumptions about the "true" stimulus  $\mathbf{M}$  underlying a sensory measurement  $\mathbf{S}$  as indicated in the previous section, we implicitly define a data model requiring a single localized activity peak of a certain size. This fits well the perceptual decision making scenario of Fig.1 and is roughly realized by the "default" lateral connectivity often used with DNFs, that is to say, Mexican hat interaction kernels with added global inhibition. This data model gives immediately rise to notions of stimulus confidence, leading to the simple concepts of conflict/ambiguity or lack of evidence as illustrated in Fig. 2.

However, in order to theoretically verify the correctness decision making using corrupted stimuli  $\mathbf{S}$  using Bayesian inference techniques, we require an explicit probabilistic model  $P(\mathbf{M}|\mathbf{S})$ , relating  $\mathbf{S}$  and  $\mathbf{M}$ , the "true" underlying stimulus. This amounts to specifying the probability of a certain  $\mathbf{M}$  being present given its corrupted measurement  $\mathbf{S}$ . Such a data model is a highly useful tool, if available, since it can be used for the following purposes:

- it defines sets of allowed or forbidden stimulus values  $\mathcal{M}^+, \mathcal{M}^-$  characterized by  $p(\mathbf{M}|\mathbf{S}) > 0 \forall \mathbf{M} \in \mathcal{M}^+$  and  $p(\mathbf{M}|\mathbf{S}) \equiv 0 \forall \mathbf{M} \in \mathcal{M}^-$
- it allows to estimate the most probable "true" stimulus  $\mathbf{M}^*$  given a corrupted stimulus  $\mathbf{S}$  as  $\mathbf{M}^* = \operatorname{argmax}_{\mathbf{M}} p(\mathbf{M}|\mathbf{S})$
- it allows to assign a confidence  $c = P(\mathbf{M}^*|\mathbf{S})$  to the most probable true stimulus. It is this confidence that should be transmitted to a subsequent hierarchy stage

In this article, we will consider only stimuli of a very simple nature, both concerning  $\mathbf{M}$  and  $\mathbf{S}$  as outlined in Sec. 2.3. In line with our definitions of ambiguity, conflict and lack of evidence as illustrated in Fig. 2, the data model should have the following simple properties:

- **punish ambiguity or conflict:** Ideally, only one dominant Gaussian should be present in the input. Confidence should decrease if the second Gaussian has a nonzero peak value
- **punish lack of evidence.** Ideally, this single Gaussian should have a peak value of 1.0. If it is lower, the confidence should decrease.

**Table 1** Parameters used in the simulation experiments

parameter	value	function	parameter	value	function
$\tau$	15	time constant	$u_{\min}$	-2	min. field potential.
$\alpha$	1	input strength	$u_{\max}$	3	max. field potential
$\beta$	4	lat. int. strength	$\theta$	0.5	transfer func. thresh
$\gamma$	0.005	noise strength	$\nu$	2.5	transfer func. slope
$h$	-1.0	resting potential	$\sigma_{\text{on}}$	6	inhibition radius
$a_0$	1	on filter strength	$b_0$	3	off filter strength
$c_0$	0.10	global inhibition	$\sigma_{\text{on}}$	3	exc. radius
$\sigma_{\text{off}}$	6	inh. radius	$N \times N$	32	neurons
$T$	280	iterations/pattern			

- **punish lack of match.** If the dominant Gaussian of the stimulus  $\mathbf{S}$  is different from the peak in  $\mathbf{M}$ , probability should drop sharply

The precise form of the model is not important for decision making as long as those properties are fulfilled. A simple choice of model is, for example:

$$P(\mathbf{S}|\mathbf{M}) \sim \exp\left(-\frac{|A_1 - M_1| + |A_2 - M_2|}{\sigma}\right) \text{ with } \sigma = 0.2 \quad (3)$$

By assuming that all stimuli  $\mathbf{M}$  are equally probable *a priori*,  $P(\mathbf{M}) = \text{const.}$ , we obtain an a posteriori probability distribution representing the sought-for data model:

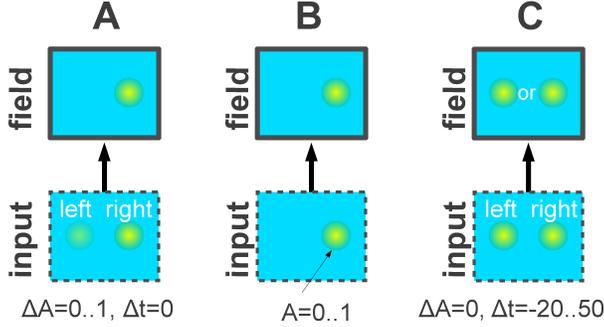
$$P(\mathbf{M}|\mathbf{S}) = P(\mathbf{S}|\mathbf{M}) \quad (4)$$

When performing decision making experiments in Sec. 3, this model will be used for verifying the correctness of the decisions. The fundamental assumption of this article is that neural field layers with a correct choice of parameters and lateral connections can approximate such a model by their dynamics. In particular, we assume that the final converged state of each layer represents the MAP estimate of this layer’s inputs, and that the response latency of this converged state encodes its confidence under the data model.

The experiments of the following section will show that this is the case at least for the simple stimuli used here.

### 3 Experiments

In all experiments, several input patterns (see Fig. 3) are presented to the network(s) for  $T$  iterations, in which time the network dynamics are simulated according to eqn. (2). Before each pattern presentation, all network potentials were reset to the value of the resting potential  $h$ . Unless stated explicitly, we use the parametrization indicated in Tab. 1 in all experiments of this section. These parameters ensure that neural layers converge to single-peak solutions and are thus in accordance with the analytical model of Sec. 2.4, see [26] for a theoretical justification. The choice of good parameters for DNFs is slightly



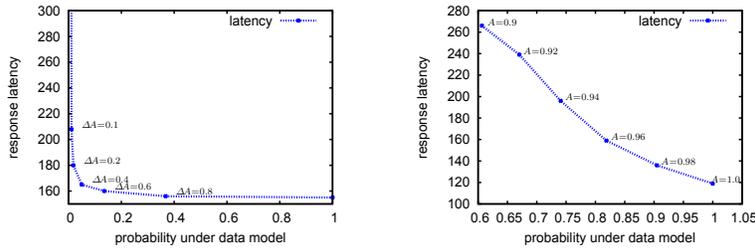
**Fig. 3** Variable input stimuli used in our experiments (lower row), leading to decisions (upper row) of variable confidence expressed by response latency. A) Variation of conflict/ambiguity, modelled by the difference  $\Delta A$  of the peak amplitudes of the left and right Gaussian. B) Variation in the degrees of evidence, modelled by the peak amplitude  $A$  of the single Gaussian stimulus C) Variation of relative latency  $\Delta t$  for a composed stimulus. In this case, each of the Gaussians has the same amplitude  $A_1 = A_2 = 1$ , whereas their relative latency is varied, thus giving each Gaussian a different influence on the dynamics of the neural layer.

tricky, especially if we wish to observe latency effects. However we have good reasons to believe that most of the parameters can in principle be determined from data using self-adaptation processes as described, e.g., in [24]. There were a few principles that we found useful in guiding our choice of parameters:

- No lateral interactions in the resting state: This implies that  $f(h) \approx 0$ , and thus constraints on  $\nu$  and  $\theta$  are introduced
- Potential cut-off must not introduce new effects. This implies that  $f(u_{\max}) \approx 1$  and  $f(u_{\min}) \approx 0$  which can be obtained by a proper setting of  $u_{\min}, u_{\max}$ .
- Lateral and afferent inputs to any neuron should be, on average, of similar magnitude. This mainly constrains  $\alpha, \beta$  and  $\gamma$ .

### 3.1 Encoding ambiguity/conflict into response latency

In the first experiment, we will demonstrate that a proper parametrization of neural layers can achieve an unambiguous translation of ambiguity/conflict (see Fig. 2) into response latency. To show this, we will successively apply stimuli of varying degrees of conflict/ambiguity to the neural layer and measure the response latency of the winning peak. Using input data according to Sec. 2.3 and Fig. 3 A), we create two Gaussian stimuli with initial amplitudes  $A_1 = 1, A_2 = 0$ , where the "true" solution is supposed to be "left", i.e.,  $M_1 = 1, M_2 = 0$ . In successive steps we increment the amplitude  $A_2$  such that the difference in amplitudes,  $\Delta A$ , goes from its initial value of 1.0 to 0.0, which always leads to a "left" solution except for  $\Delta A = 0$ . Evidently, this variation of  $A_2$  reduces the probability  $P(\text{"left"}|\mathbf{S})$  of the attractor solution under the probabilistic model of. (3). As shown in Fig. 4 (left), we observe a monotonous



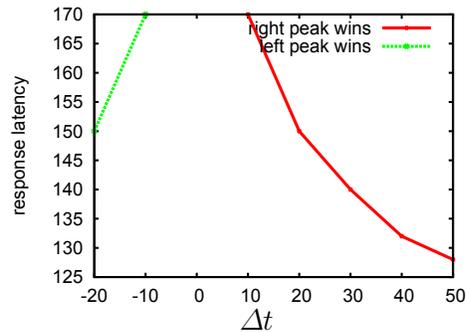
**Fig. 4** Encoding of decision confidence using single neural layers for input stimuli  $\mathbf{S}$  with different types of structured noise: "ambiguity/conflict" (left diagram) as well as "lack of evidence" (right diagram). Peaks always develop on the "right" position in the encoding field with an input-dependent latency, expressing the MAP estimate along with a decision confidence. Both graphs show a monotonous relationship between response latency and the probability under the analytical data model,  $P(\text{"right"}|\mathbf{S})$ , underlining the relatedness of those quantities. For each measurement point in both graphs, the associated parameters of the input stimulus are given as well: either the difference between conflicting peak amplitudes  $\Delta A$ , or the strength of the single peak  $A$ .

but nonlinear relationship between  $P(\text{"left"}|\mathbf{S})$  and response latency, showing that the latter can indeed be used unambiguously to express the former.

### 3.2 Encoding lack of evidence into response latency

Similarly to the previous section, we create input stimuli according to Sec. 2.3 and Fig. 3B), although now with only one Gaussian "left" stimulus whose amplitude  $A$  is varied from 1.0 to 0.9, thus reducing confidence of the attractor solution  $P(\text{"left"}|\mathbf{S})$  computed using eqn. (3). The "true" stimulus is supposed to be "left", i.e.,  $M_1 = 1, M_0 = 0$ . As can be observed in Fig. 4 (right), there is a monotonous, almost linear relation between  $P(\text{"left"}|\mathbf{S})$  and response latency. This result shows that what we called "lack of evidence" in Sec. 1 reduces the confidence of decisions under the analytic data model, and that this confidence is unambiguously represented by response latency.

The reason we keep the amplitude of the Gaussian in the range of  $[0.9, 1.0]$  is that the latencies grow excessively for  $A \rightarrow 0.9$ , up to the point where no activation is produced at all within  $T$  iterations for  $A < 0.9$ . While it is straightforward to find neural layer parameters that allow the representation of  $A < 0.9$ , they are usually not compatible with the encoding of uncertainty/ambiguity demonstrated in Sec. 3.1 when the full range of uncertainty/ambiguity should be represented. This seems to be a constraint of the recurrent networks model, see Sec. 4. As this article focuses more on the representation of ambiguity/conflict, we demonstrate the encoding of "lack of evidence" for completeness, but will not use it further in what follows.

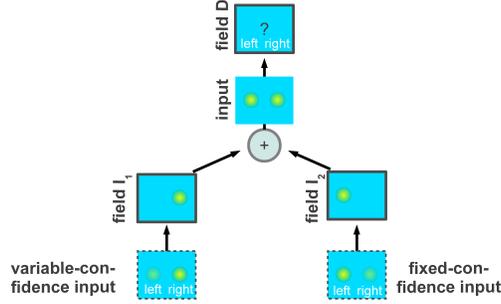


**Fig. 5** Decoding response latency. The experiment considers a single neural layer receiving an input composed of two equally strong Gaussian stimuli (denoted "left" and "right") with relative latency  $\Delta t$ . Depending on the  $\Delta t$ , a peak forms either at the "left" or the "right" position (green/red curves), with a latency depending on  $\Delta t$  which can be seen in the diagram. For  $\Delta t = 0$ , no peak forms during the observation interval, corresponding to the refusal to take an impossible decision.

### 3.3 Decoding response latency

After having shown that decision confidence is translated unambiguously into response latency, we are now going to show that latency can be decoded to influence decision-making in just the right way. For this purpose we will present a neural layer with an input containing two Gaussians of equal peak amplitude  $A_1 = A_2 = 1$ , arriving with different latencies  $t_1, t_2$ . We assume the "true" stimulus to reside at the location of the earlier-arriving Gaussian. This is intended to emulate the situation where these two Gaussians come from two separate neural layers faced with inputs of different confidence, see Fig. 3D). As a two-peak input does not fit the implicit data model of the neural layer, it is forced to take a decision, and we want to find out how this decision is affected by the relative latency  $\Delta t$ .

As can be observed in fig. 5, we find that  $\Delta t$  has a strong influence on decision making. Not only does the first-arriving Gaussian completely dominate the layer's response, but we also find that, as  $\Delta t$  decreases, response latency increases monotonously. Both findings are intuitive in the sense of optimal decision making: as the latency of inputs can be linked to the confidence of the decisions that generated them (see Sec. 3.1), a good decision must be in favour of the earlier (i.e., more confident) input. Likewise, if two inputs have similar latencies (i.e., confidences), the confidence of the decision should be lowered, which is expressed by an increase in latency. This supports the construction of deeper hierarchies, since the decision of the decoding field has itself a confidence that is again expressed as response latency.



**Fig. 6** Schematics of the three-stage hierarchy of recurrent neural layers used for the testing of optimal decision making. The variable noise in the input to  $I_1$ , and the fixed noise in the input to  $I_2$  are translated into variable and fixed latencies of neural responses in  $I_1$  and  $I_2$ . For simplicity, stimuli are chosen such that the "right" stimulus always wins the competition in  $I_1$ , whereas the "left" stimulus always wins in  $I_2$ . This leads to a competition between those two positions in  $D$ , which decides between "left" and "right" based on the relative latency of inputs from  $I_1$  and  $I_2$ .

### 3.4 Optimal decision making in a hierarchy

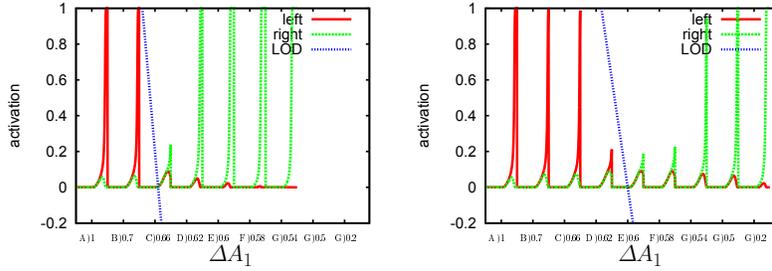
After the preparations of the previous subsections, we are now ready to construct a 3-stage hierarchy out of identical neural layers. The hierarchy is constructed as shown in Fig. 6, and all layers use the parameters of Tab. 1. Two low-level layers,  $I_1$  and  $I_2$ , encode the confidence (which is varied for  $I_1$ ) of their inputs into response latency, which is in turn decoded by the field  $D$ .

As we know that the implicit data model of neural fields, encoded into their connectivity and dynamics, can approximate the analytical data model of Sec. 2.4 as shown in Secs. 3.1, 3.2, 3.3, this experiment is meant to determine whether a deeper hierarchy will respect the analytical data model as well, amounting to optimal decision making behavior in  $D$  for arbitrary decision confidences in  $I_1$  and  $I_2$ . Inputs to both  $I_1$  and  $I_2$  are of the "conflict/ambiguity" type illustrated in Fig. 3A): Whereas the confidence in  $I_2$  is kept constant by using a constant difference between peak amplitudes  $\Delta A_2 = 0.6$ , the confidence in  $I_1$  is varied by manipulating the amplitude difference  $\Delta A_1$  in analogy to in Sec. 3.1.

To determine the optimal decision to be taken by  $D$ , we need to calculate the a posteriori distribution  $P(\mathbf{M}|\mathbf{S}_1\mathbf{S}_2)$ . Assuming independence of  $\mathbf{S}_1$  and  $\mathbf{S}_2$ , we can simply multiply and obtain

$$P(\mathbf{M}|\mathbf{S}_1\mathbf{S}_2) = P(\mathbf{M}|\mathbf{S}_1)P(\mathbf{M}|\mathbf{S}_2) \quad (5)$$

To obtain a decision, we must compare the a posteriori probabilities  $P(\text{"left"}|\mathbf{S}_1\mathbf{S}_2)$ ,  $P(\text{"right"}|\mathbf{S}_1\mathbf{S}_2)$ . To this effect, we calculate the sign of the log-odds-difference



**Fig. 7** Optimal decision making in a three-stage hierarchy, see also Fig. 6 and text. Low-level decisions and their confidences, arising from a variable "ambiguity/conflict" type of input (parametrized by  $\Delta A$ , the difference between peak amplitudes) to the lower-level field  $I_1$ , are decoded by the high-level field  $D$  to take a Bayes-optimal decision. Shown is the temporal development of neural activities in  $D$  at the positions of the "right" (red graphs) and "left" (green graphs) stimuli. The blue line indicates the theoretical log-odds-difference which indicates the optimal decision ("left"/"right") by its sign. Left: coarse resolution covering the complete range of  $\Delta A$ . Right: finer resolution around the pivotal point of  $\Delta A = 0.6$ .

(LOD):

$$\begin{aligned}
 \text{sgn}(\text{LOD}) &= \text{sgn}\left(\log \frac{P(\text{left}|\mathbf{S}_1\mathbf{S}_2)}{P(\text{right}|\mathbf{S}_1\mathbf{S}_2)}\right) \\
 &= \text{sgn}(\log P(\text{left}|\mathbf{S}_1\mathbf{S}_2) - \log P(\text{right}|\mathbf{S}_1\mathbf{S}_2)) = \\
 &= \text{sgn}(6 - 10\Delta A_1).
 \end{aligned} \tag{6}$$

Given the nature of the inputs  $\mathbf{S}_1, \mathbf{S}_2$  as depicted in Fig. 6,  $D$  should decide "right" if  $\text{sgn}(\text{LOD}) = 1$ , it should not decide if this quantity is zero, and it should decide "left" when it is  $-1$ . When viewing the results given in Fig. 7, the decision behavior exhibited by  $D$  exactly matches the optimal decisions indicated by the LOD calculation. We observe that, for  $\Delta A_1 > \Delta A_2$ , the "right" stimulus from  $I_1$  wins the competition in  $D$  as it is more confident; similarly, the "left" stimulus from  $I_2$  wins for  $\Delta A_1 < \Delta A_2 = 0.6$ . At  $\Delta A_1 = \Delta A_2$ , output in  $I_1$  and  $I_2$  have equal confidence, reflected in a refusal of  $D$  to take a decision (no significant activity develops either at the "left" or "right" position).

### 3.5 Resolution of decision making in a hierarchy

As neural responses are usually quite noisy, this will obviously affect a neural code using response latency. We therefore wish to know how the (weak) noise we simulate in our neural layers affects decision making when taking near-ambiguous decisions. As shown in Fig. 7 (right), decision making is still optimal in cases where amplitude differences in  $I_1$  and  $I_2$  do not get too similar. An exception are cases where amplitude differences are within 0.02 of

each other (e.g., case D) in Fig. 7) (right): here the resolution of response latency is apparently insufficient to encode and decode confidence with sufficient precision. For these borderline cases, the behavior of  $D$  has a strong random component, that is to say, the decision taken by  $D$  depends on the precise form and the initial conditions of the noise.

## 4 Discussion

In this section, we will first review the results that were obtained, then point out and discuss topics that merit special attention, as well as list shortcomings and limitations of our work. As a last point, we will try to generalize the findings of this article, and conclude with an outlook of future work.

### 4.1 Summary

We presented experiments that perform statistically optimal decision making in a 3-layer hierarchy of recurrent neural layers, implemented as dynamic neural fields. It was shown that, as a consequence of the non-linear recurrent dynamics in each layer, conflict/ambiguity or lack of evidence in input stimuli can be unambiguously encoded as response latency of attractor states. Similarly, response latency can be decoded by subsequent hierarchy stages, taking into account the confidence of lower-level decisions. Finally, it was verified by comparison to an exact probabilistic model that a hierarchy of neural fields can indeed take Bayes-optimal decisions by exploiting the information contained in response latency, and that this ability is robust to moderate noise levels.

### 4.2 Particular points

*Parametrization* All of the aforementioned functions were realized by finding appropriate parametrizations for the DNF model, not by making changes to the model itself. It is important to note that the encoding and decoding of latency are inherent in the neural dynamics when correct parameters are chosen. Clearly, the parameters of different neural layers in a hierarchy need not be identical; however it is imperative that the chosen parameters allow each neural layer to encode and decode latency within the simulation time for a single stimulus,  $T$ . Fortunately, using the guidelines for parametrization mentioned in Sec. 3, we were able to find a single set of parameters appropriate for all hierarchy layers. That encoding and decoding can be performed with the same parameter settings is not self-evident, but we found, in the course of numerous experiments, that parameters allowing encoding were almost always suited for decoding as well. Since the process of finding the required parameter values is not straightforward at all, we believe self-adaptation processes[35,24] must be applied to automate the process of adapting parameters to data statistics.

Since latency encoding and decoding were found to work for any parametrization where input stimuli could create an attractor in less than  $T$  iterations, self-adaptation processes would just have to ensure that this condition is met, e.g., by maintaining temporal averages of neural activations over single pattern presentation ( $T$  iterations) at some target value. If no activations appear within  $T$  iterations, such averages would be too low, while they would be too high if neurons were constantly active over the whole period  $[0, T]$ .

*Representation of probability* In the presented study, we effectively assume that recurrent neural dynamics approximate a probabilistic data model  $P(\mathbf{M}|\mathbf{S})$  for the "true" values of input variable  $\mathbf{M}$  given the afferent inputs  $\mathbf{S}$  which are afflicted by (structured) noise. At least for simple stimuli, we have shown that our recurrent network approach represents both the MAP estimate  $\mathbf{M}^* = \arg \max P(\mathbf{M}|\mathbf{S})$  (represented by the position of the localized attractor solution), as well as a nonlinear transformation of  $P(\mathbf{M}^*|\mathbf{S})$  itself (represented by response latency). As can be seen from the results of Sec. 3.1, this encoding of the decision confidence  $P(\mathbf{M}^*|\mathbf{S})$  is one-to-one but non-linear, so that response latency does not directly represent a probability here. This is however not required to take correct decisions: as all we need to do is to *compare* latencies in different areas of the input while assuming they were generated using the same underlying data model. In this study, the model  $P(\mathbf{M}|\mathbf{S})$  is effectively unimodal because a single Gaussian is assigned the highest a posteriori probability. This is reflected in the lateral connectivity of the used networks which locally favors Gaussian activation peaks but restricts their number to one due to global inhibition. There is however nothing whatsoever to keep us from implementing or learning other, more general data models by adapting the lateral connectivity structure of recurrent networks.

*Comparison to related work* When considering the representation of probability we discussed in the last paragraph, several important differences to related work are notable: First of all, and different from [15, 29, 8], our approach does not treat neural activities as log-probabilities. More generally, and in line with [23], we do not treat the set of input activations  $\mathbf{S}$  as a probability distribution but as a collection of physical measurements corrupted by noise. In contrast to [23], however, we do not require this noise to have a particular form as long we can represent it in our internal data model  $P(\mathbf{M}|\mathbf{S})$ , encoded in the lateral connections of each layer. A further difference to [23], who consider Bayesian integration separately for each neuron, is that we consider data models that are global in the sense that they consider the values of other, not necessarily adjacent neurons in the computation of the a posteriori probability. Summarizing, this article suggests a new way of approximately representing and processing probabilistic information in neural hierarchies which is quite different from what has been proposed in previous works, although it has been validated only for very simple stimuli. To be fair, on the other hand, most previous work on the subject uses test stimuli of similar simplicity with the possible exception of [29].

### 4.3 Limitations

*Noise* In the simulations of Sec. 3, noise was kept at an intermediate level, and we could show that the "resolution" for stimuli whose confidence was similar was in fact quite good, see Sec. 3.5. Experiments conducted with higher levels of noise show that stimuli of similar confidence can interchange their temporal order as a consequence of noise. To react to this loss of resolution, the parameters of neural layers must be adapted such that the range of confidences that can be represented is narrowed, giving more resolution to the high confidences and dropping the lower ones. At the same time, the mechanisms underlying attractor formation could be revisited and adapted, thus making them less susceptible to noise. Clearly, the behavior of our approach under stronger noise merits further attention.

*Can every probabilistic model be approximated?* In the analysis of the experiment of Sec. 3.2, it was mentioned that we were unable to find a set of parameters for which the neural layers could represent the full range of confidences arising from either lack of evidence and ambiguity/conflict. Maybe such parameters exist, but we believe it is more likely that not every conceivable model may be approximated faithfully by recurrent neural dynamics. What can be done, however, is to restrict the range of represented confidences both for conflict/ambiguity and lack of evidence to a point where both can be represented faithfully, at the cost of not representing some of the less confident patterns. We do not consider this to be a significant problem since only the least confident inputs would have to be disregarded (who would not win subsequent competitions in any case). Nevertheless it will be worth investigating whether, and how, the greatest possible set of sensible data models can be approximated by recurrent dynamics.

*Loss of information in deeper hierarchies* The basic creed of Bayesian data processing[4] is to always transmit and manipulate distributions, and to take the final decisions only at the end of a processing chain by choosing the argmax of the final distribution. This is clearly a very intelligent strategy since no information is lost on the way through, e.g., a decision hierarchy. Thus, if apparently irrelevant alternatives are not already disregarded at low hierarchy stages, they might still change the final decision when integrated at higher stages. The presented study outlines a possible way towards this goal, for although competing stimuli do get suppressed at the lowest hierarchy stages, see Sec. 3.4, at least the fact that something significant was suppressed is transmitted as confidence, and taken into account at the next stage. What is more, with another data model that allows for several localized stimuli in the input, alternatives could be transmitted through the hierarchy. Nevertheless, each hierarchy stage potentially suppresses information using its data model, and thus the full "Bayesian dream" is not yet realized by our approach. It can (and should) of course be debated whether a full Bayesian treatment is really required for robust real-world processing, or whether the chosen approximation

will already be feasible or even beneficial, as transporting the full probabilistic information might be strongly susceptible to noise as well.

*Generality* The results presented here were obtained using the dynamic neural field model as a basis for simulation, so we wish to discuss how our results will generalize to other recurrent network models, especially spiking ones. This of course depends to a great extent on the precise form of the used model equations, but at least for spiking models using an integrate-and-fire type of update equation we can be reasonably sure that our findings will hold true. As response latency is due to delayed build-up of membrane potential (caused by lateral inhibition), and since the basic update equation for the membrane potential in integrate-and-fire models is identical to eqn. (1) except for spike generation and reception, the same mechanism should apply, although of course with different parameters. This assumption is supported by reports of response latency in [33] with integrate-and-fire models.

## 5 Outlook and future work

As we already mentioned the previous section, it is for the most part the lateral connection weights of a recurrent neural layer that encode the data model  $P(\mathbf{M}|\mathbf{S})$ . In the present study, this model was very naive in just admitting a single Gaussian activity peak as an attractor solution which we assume to represent the MAP estimate  $M^* = \arg \max P(\mathbf{M}|\mathbf{S})$ . The advantage of using a neural network is that lateral connection weights, and thus an internal data model, could be *learned from data statistics* by simple correlation-based rules. On the minus side, this would require giving up the notion of a location-independent convolution kernel in eqn.(1), resulting in much higher memory consumption. Nevertheless, this would allow to model real-world inputs coming, e.g., from a visual sensor, and to use the internal data model to detect deviations from the ordinary for more robust processing. This will be especially relevant for our work on autonomous and self-organized learning in real-world scenarios (see, e.g., [14]), where it must be determined whether complex new representations of visual stimuli are out of the ordinary, and should thus trigger learning. The detail of such a combined learning process will certainly be the subject of future work, as well as the question of how to self-adapt the other parameters of a recurrent network to best match the data statistics. Furthermore, the influence of noise on the latency code will have to be investigated with the goal of allowing optimal decision making even in the face of strong noise. All these efforts will lead to a better understanding of how the human brain processes probabilistic information, with the aim of creating more robust intelligent systems operating under real-world conditions.

## References

1. S.-i. Amari. Mathematical foundations of neurocomputing. *Proceedings of the IEEE*, 78(9):1441–1463, 1990.
2. Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle. Greedy layer-wise training of deep networks. *Advances in neural information processing systems*, 19:153, 2007.
3. E. Bicho, L. Louro, and W. Erlhagen. Frontiers. *Frontiers in Neurobotics*, 4, 2010.
4. C. Bishop. *Pattern recognition and machine learning*. Springer-Verlag, New York, 2006.
5. R. Borowsky and M. Masson. Semantic ambiguity effects in word identification. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 22(1):63, 1996.
6. P. Cisek. Integrated neural processes for defining potential actions and deciding between them: a computational model. *J Neurosci*, 26(38):9761–9770, Sep 2006.
7. P. Cisek. Cortical mechanisms of action selection: the affordance competition hypothesis. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 362(1485):1585–1599, 2007.
8. R. Cuijpers and W. Erlhagen. Implementing bayes’ rule with neural fields. In *ICANN ’08, Part II*, pages 228–237, Berlin, Heidelberg, 2008. Springer-Verlag.
9. G. Deco and E. T. Rolls. A neurodynamical cortical model of visual attention and invariant object recognition. *Vision Res*, 44(6):621–642, Mar 2004.
10. S. Deneve and A. Pouget. Basis functions for object-centered representations. *Neuron*, 2003.
11. W. Erlhagen and G. Schöner. Dynamic field theory of movement preparation. *Psychological review*, 109(3):545, 2002.
12. C. Faubel and G. Schöner. Learning to recognize objects on the fly: a neurally based dynamic field approach. *Neural Networks Special Issue on Neuroscience and Robotics*, 21(4):Pages 562–576, May 2008.
13. J. Gautrais and S. Thorpe. Rate coding versus temporal order coding: a theoretical approach. *Biosystems*, 48(1-3):57–65, 1998.
14. A. Geppert. Efficient online bootstrapping of representations. *Neural Networks*, 2012.
15. J. Gold and M. Shadlen. Neural computations that underlie decisions about sensory stimuli. *Trends Cogn Sci*, 5(1):10–16, Jan 2001.
16. E. Hazeltine, R. Poldrack, and J. Gabrieli. Neural activation during response competition. *Journal of Cognitive Neuroscience, Supplement 2*, pages 118–129, 2000.
17. G. Hinton, S. Osindero, and Y. Teh. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554, 2006.
18. J. S. Johnson, J. P. Spencer, and G. Schöner. Moving to higher ground: The dynamic field theory and the dynamics of visual cognition. *New Ideas Psychol*, 26Kamps(2):227–251, Aug 2008.
19. R. Kiani, H. Esteky, and K. Tanaka. Differences in onset latency of macaque inferotemporal neural responses to primate and non-primate faces. *J Neurophysiol*, 94(2):1587–1596, Aug 2005.
20. D. C. Knill and A. Pouget. The bayesian brain: the role of uncertainty in neural coding and computation. *Trends Neurosci*, 27(12):712–719, Dec 2004.
21. R. Kupper, M.-O. Gewaltig, U. Körner, and E. Körner. Spike-latency codes and the effect of saccades. *Neurocomp.*, 65-66:189–194, 2005. Special issue: Computational Neuroscience: Trends in Research 2005 – Edited by E. de Schutter.
22. Y. LeCun, F.-J. Huang, and L. Bottou. Learning methods for generic object recognition with invariance to pose and lighting. In *Proceedings of CVPR’04*. IEEE Press, 2004.
23. W. Ma, J. Beck, P. Latham, and A. Pouget. Bayesian inference with probabilistic population codes. *Nature Neuroscience*, 9(11), 2006.
24. A. M.Garcia Ortiz. Neural self-adaptation for large-scale system building. In *First International Conference on Cognitive Neurodynamics*, 2009.
25. T. Michelet, G. Duncan, and P. Cisek. Response competition in the primary motor cortex: Corticospinal excitability reflects response replacement during simple decisions. *Journal of Neurophysiology*, 104(1), 2010.
26. I. Mikhailova and C. Goerick. Conditions of activity bubble uniqueness in dynamic neural fields. *Biol Cybern*, 92(2):82–91, Feb 2005.

27. M. W. Oram, D. Xiao, B. Dritschel, and K. R. Payne. The temporal resolution of neural codes: does response latency have a unique role? *Philos Trans R Soc Lond B Biol Sci*, 357(1424):987–1001, Aug 2002.
28. A. Pouget, S. Deneve, and J.-R. Duhamel. A computational perspective on the neural basis of multisensory spatial representations. *Nat Rev Neurosci*, 3(9):741–747, Sep 2002.
29. R. P. N. Rao. Bayesian computation in recurrent neural circuits. *Neural Comput*, 16(1):1–38, Jan 2004.
30. D. S. Reich, F. Mechler, and J. D. Victor. Temporal coding of contrast in primary visual cortex: when, what, and why. *J Neurophysiol*, 85(3):1039–1050, Mar 2001.
31. N. P. Rougier and J. Vitay. Emergence of attention within a neural population. *Neural Netw*, 19(5):573–581, Jun 2006.
32. Y. Sandamirskaya, J. Lipinski, and I. Iossifidis. Natural human-robot interaction through spatial language: a dynamic neural fields approach. In *Proc. 19th IEEE International Workshop on Robot and Human Interactive Communication (ROMAN 2010)*, 2010.
33. S. Schrader, M.-O. Gewaltig, U. Körner, and E. Körner. Cortext: A columnar model of bottom-up and top-down processing in the neocortex. *Neural Networks*, 22(8):1055 – 1070, 2009.
34. J. Taylor. Neural ‘bubble’ dynamics in two dimensions: foundations. *Biological Cybernetics*, 80:393–409, 1999.
35. G. G. Turrigiano and S. B. Nelson. Homeostatic plasticity in the developing nervous system. *Nat Rev Neurosci*, 5(2):97–107, Feb 2004.
36. R. Van Rullen, J. Gautrais, A. Delorme, and S. Thorpe. Face processing using one spike per neurone. *Biosystems*, 48(1-3):229–239, 1998.
37. C. Wilimzig, S. Schneider, and G. Schneider. The time course of saccadic decision making: dynamic field theory. *Neural Netw*, 19(8):1059–1074, Oct 2006.
38. R. S. Zemel, P. Dayan, and A. Pouget. Probabilistic interpretation of population codes. *Neural Comput*, 10(2):403–430, Feb 1998.
39. S. K. U. Zibner, C. Faubel, J. P. Spencer, I. Iossifidis, and G. Schöner. Scenes and tracking with dynamic neural fields: How to update a robotic scene representation. In *Proc. Int. Conf. on Development and Learning (ICDL10)*, 2010.