A model of the binocular rivalry condition based on a hierarchy of conceptor-controlled neural networks

Felix Meyer zu Driehausen [1,*], **Johannes Leugering** [1] **and Gordon Pipa** [1]

[1] *Institute of Cognitive Science, University of Osnabrück, 49069, Germany*

Correspondence*:
Felix Meyer zu Driehausen
Institute of Cognitive Science, University of Osnabrück, Wachsbleiche 27, 49069
Osnabrück, Germany, fmeyerzudrie@uni-osnabrueck.de

ABSTRACT

Binocular rivalry and other forms of bistable perception have been researched intensively within the last century. Besides these long standing efforts many questions remain. It becomes apparent that rivalry involves multiple distributed processes and that several layers in the hierarchy of sensory processing are involved. This observation directs research towards general models of perceptual inference and to the question whether rivalry stimuli can be rooted within these models. This thesis attempts such a synthesis. We chose a recent explanation of binocular rivalry in terms of predictive coding as a departure point. In order to instantiate this theoretical framework we implemented a sensory processing hierarchy consisting of three layers of conceptor controlled recurrent neural networks. The perception of this system was observed, while it was exposed to a mixture of two signals that were learned beforehand and it was compared to acknowledged results from research on binocular rivalry in humans. This research contributes threefold. (1) We build a pioneering computational model on a very general perceptual framework that can account for the particularities of bistable perception. (2) While the effects of the binocular rivarly stimuli onto the system were not specifically engineered, the ability of the system to seamlessly integrate these phenomena is evidence for the framework theory of predictive coding. (3) Last but not least, conceptor controlled recurrent neural networks are shown to be suitable to implement the particular condition of binocular rivalry, suggesting that their usage will be fruitful in a wide variety of cognitive modelling applications.

Keywords: Bistable perception, binocular rivalry, reservoir computing, echo state networks, conceptor, predictive coding

1 INTRODUCTION

Human perception is arguably one of the most fascinating subjects of research. During the long history of investigation of the humans' ”window into the world” many perceptual phenomena that seem unusual or erroneous caught attention of the researchers. Among these are oddities and irregularities such as illusions and bistable perception. These have often been challenging for existing theories to accommodate, but at the same time served as hints to insight into the inner workings of the perceptual system. In the domain of vision, the phenomenon of binocular rivalry, a form of bistable perception, has been studied intensively. Binocular rivalry occurs when two different and conflicting stimuli are presented separately to each eye. A well known example of such stimuli are pictures of houses and faces. Besides being different they are moreover incompatible, because the compound ”house-face” or rather loosely said a ”house that looks like a face” or a ”face that looks like a house” are highly unlikely to be encountered in daily human life. A binocular rivalry condition could be set up by presenting a house to one eye and a face to the other. Instead of perceiving the unlikely compound of both, the subject perceives them separately and in an alternating manner. While one stimulus is consciously perceived, the other is suppressed. As the phenomenon of binocular rivalry has been studied intensively, there are also computational models of binocular rivalry that can account for many of the observations from psychophysical experiments. Anyhow, most of these models do not state a general framework for perceptual inference wherein the observed effects of the binocular rivalry condition fall into place, but are rather specifically build for the binocular rivalry condition. There is no question that these models proved to be useful to gain insights into perception. But if one adopts the view that at the basis of the perceptual system there is one general working mechanism, the phenomenology of the special case of binocular rivalry should naturally be accounted for by the general perception algorithm, given the special input signal of a combination of incompatible stimuli.

The aim of this thesis is to develop a computational model which experiences a perception similar to humans in the binocular rivalry condition. Most importantly, the structure and working algorithm of the model are based on a general framework for perception. But what is this kind of structure and the general algorithm for perception? To our knowledge there exist only few attempts to explain binocular rivalry within a framework for general perception. One such approach is given by Hohwy et al. (2008), which also serves as a starting point for this work. It utilizes predictive coding theory or predictive error minimization theory (PEM) to explain the phenomenology of binocular rivalry. PEM is a general theory for human perception and moreover claims to *unify* action, perception and attention. As the predictive coding theory for the brain became more popular only recently, we will introduce the general concept here and furthermore provide more detail in the following chapters.

The central ingredient of a system that operates according to the predictive error minimization principle is a generative hierarchical model. The need for this model in the human brain becomes apparent by the epistemic constraints that our brain is subject to, being inside the skull and only having indirect access to the world through varying sensory signals Clark (2013). In other words the brain never is in direct contact with the world but can only take measurements with the sense organs and extract regularities from this data. How can it know something about the world if it only has access to the signals that are the effects of the causes in the real world? This is the challenging task of perception, to infer the causes on the sole basis of the effects. According to PEM the brain meets this challenge by maintaining a generative hierarchical model of the world. This model continuously generates hypotheses of the worldly causes. In employing this mechanism the brain escapes the tricky task of reconstructing the chain of causes and effects backwards. Now it has access to two quantities of effects, those that are self generated by the generative hierarchical model and those that are input through the sense organs. In an optimal case these should be similar, this would give the system confidence that the current hypothesis is well suited to explain the incoming sensory data. It is reasonable to interpret the difference between the self generated signal and input signal as prediction error. This error is used as feedback on the internal model of the world. Minimizing the prediction error then translates to maximizing the accuracy of the internal models of the world. In line with this is the central claim of PEM: the brain's overall objective is to organize neural activity in a way that it most efficiently minimizes prediction error, on average and on multiple levels of the hierarchy. Hohwy (2013)

Reconnecting to the observations in the binocular rivalry condition, how can we phrase the phenomenology of binocular rivalry in terms of the predictive coding principle? According to predictive coding theory the brain tries to find the best matching hypothesis that could be the cause for the observed data. In the binocular rivalry condition two incompatible stimuli are presented. We call them incompatible, because the brain is not used to the superposition or combination of both. Humans for example are not used to see faces and houses in the same place at the same time. Therefore the hypothesis that the cause for sensory data is the superposition of a house and a face is a priori and due to past experiences highly unlikely. The hypothesis that either one, the house or the face, is the cause for the sensory input, can alone only explain about half of the observed data. When the human brain is exposed to a binocular rivalry condition with houses and faces as stimuli, it settles for example on the hypothesis that a house caused the visual stimulation. Under this hypothesis, the brain as a hierarchical generative model would predict some features of a house which will match with parts of the sensory data. Anyhow, the sensory drive that is generated by the face would remain as a residuum and as a prediction error that is not accounted for by the prediction of the brain. This error is on about the same order of magnitude as the explained data, namely the part of the stimulus that belongs to the house. Due to this balance of information content between both parts of the stimulus, at a certain point the hypothesis that the face generated the sensory drive would overtake. This oscillation describes the alternation between different percepts that is observed when humans view rivalling stimuli.

This explanation of the binocular rivalry condition is so appealing, because it falls easily into place in the framework of predictive coding. It makes intuitively sense, but so far it is only an explanation under a hypothesis, which could use some support or evidence. We believe that a biologically inspired computational model would be fruitful in order to test this hypothesis. We have build such a model and therefore had to choose a suitable platform. Which platform is well suited to incorporate the predictive coding framework? Which is at least not biologically *implausible*?

The basis of our architecture is a reservoir computing system, in particular an echo state network Jaeger (2001). The principle part of such a system is the reservoir. In general any excitable medium that reacts to incoming drive in a non-linear way and that possesses a certain amount of memory of recent states can serve as a reservoir. In our architecture we use an assembly of randomly connected analog and non-spiking artificial neurons as a reservoir. This is the type of setup that is usually referred to as an echo state network (ESN). By being random, the connectivity of the network very likely is also cyclic. The reservoir neurons have non-linear activation functions ($tanh$) which define how they react to incoming drive. Furthermore the reservoir system possesses a short term memory because recent input signals resonate within the network due to the cyclic connectivity. Reservoir systems resemble some properties, such as recurrent connections, which are also found in biological systems. All these properties are also found in standard recurrent neural network models. In contrast to these, reservoir computing and therefore also echo state networks make use of a different learning algorithm. Instead of adapting the internal connections within the reservoir, these usually remain at their initial random values. Within the paradigm of reservoir computing only an output mapping is learned by simple linear regression. The reservoir performs a feature expansion on the input data, comprising memory and non-linearity. This expansion will subsequently, in the learning step, be linearly combined to produce a desired output. ESN's therefore can also be used to learn a generative model of a signal. The binocular rivalry condition requires the system to learn *two* distinct signals. With a bare ESN this is not possible. We use the recently by Jaeger (2014) introduced Conceptor mechanism in order to learn two patterns within the same reservoir. A conceptor, at the very end, is nothing but a regularized identity map on the reservoir state space that is included in the networks state update loop. In more intuitive words, a conceptor represents common activations of the neurons in the reservoir when this reservoir is exposed to a specific pattern. For a different pattern the network exhibits different activations. A conceptor is a filter that is learned for a specific pattern and a specific network, which suppresses all non typical network activation for the pattern and the network in question. By inserting a conceptor into the state update loop of the reservoir system, a new dynamical system with a specific attractor is formed. This system can, if everything worked well, reproduce the learned pattern. If one inserts another conceptor, a distinct dynamical system with a distinct attractor is formed, generating another pattern. This mechanism allows to learn a generative model of several patterns within a reservoir. It also allows to represent hypotheses by calculating the match of a set of conceptors to the network activity. We need exactly this property to identify a learned signal in the incoming drive to the reservoir.

We furthermore stacked three identical systems of this kind, echo state networks plus conceptors, in a hierarchy and introduced a feedback loop from the top module to the lowest module which adjusted the input to the system according to the current hypothesis. Effectively we subtracted the part of the signal from the input which is predicted under the current hypothesis, leaving only the part which could not be accounted for by the current prediction. This is the prediction error, which in case of the binocular rivalry condition is the complete signal that is not dominant at the moment. This feedback loop poses the objective to minimize prediction error onto the system, exactly as it is proposed by predictive coding theory. As the system settles on a hypothesis, it produces the corresponding signal. The part that is explained or predicted by the current hypothesis is eliminated from the incoming drive, leaving only the unexplained ”prediction error”. This error can be explained by the other hypothesis, leading to a change of hypothesis of the system and overall in an oscillation and alternating behaviour. This behaviour resembles the experiences a human observer reports when viewing rivalling stimuli.

2 MATERIAL & METHODS

Research on brains and cognition approaches cognitive phenomena on various levels of description. A common way of distinguishing cognitive phenomena is to separate them into high-level processes such as logical reasoning, planning and language on the one hand and low-level processes such as various modalities of sensory processing and motor control on the other hand. In line with this differentiation high-level phenomena are investigated in the top-down direction by use of symbolic formalisms, where in contrast low level phenomena are investigated in the bottom-up direction with the use of analytical tools such as statistics and information theory Gähde et al. (2014); Jaeger (2014). The human brain has implemented high-level cognitive functions on the basis of low-level neuro-dynamical processes and therefore has overcome the gap between raw data representation and symbolic structures. This neuro-symbolic integration problem has to a large extend remained an open question within cognitive science. A recently proposed ”Conceptor Controlled Recurrent Neural Network” is a promising biologically not implausible architecture which represents in a natural way concepts within its neural state dynamics Jaeger (2014). In the following we will provide an overview of the key properties of a conceptor controlled reservoir system.

2.1 Conceptors

So called ”Conceptors” act as filters on the trajectory of the system in state space. They are motivated by the observation that a reservoir system, when receiving a certain input, inhabits regions of the state space that are characteristic for that input. In particular this means that for different patterns it visits different regions of the state space. Conceptors for a specific pattern describe these regions in state space that a particular reservoir system visits when it is exposed to the pattern in question, or, if the pattern was learned by the reservoir system, is autonomously generated in the absence of input. Let us have a closer look at the geometrical shape that a conceptor represents. In order to learn a conceptor C_p for a pattern p a reservoir first has to be exposed to p and the network activation states x have to be recorded. Subsequently principal component analysis (PCA) on the collected states is performed. This yields singular values and eigenvectors that describe the geometry of the point cloud of states in state space. In the example of a 2-neuron reservoir the state space is two dimensional. A possible point cloud in that state space is depicted in Figure **??** in the left graph as grey dots. The eigenvectors of the point cloud span the ellipses with eigenvalues σ_1 and σ_2 corresponding to the length of the eigenvectors. In the right graph these eigenvalues were scaled and thereafter define a new, modified ellipse. The scaling is dependent of a regularization parameter α, termed *aperture* by Jaeger (2014). It negotiates between two terms of a cost function: The sum of all squared matrix entries of the conceptor matrix on the one hand and on the other hand the difference between the conceptor filtered reservoir activations and the unfiltered activations. It is a regularization parameter that compromises the degree of filtering that the conceptor matrix induces with the absolute value of the weights in the concept matrix. Reversing the PCA with the modified eigenvalues will result in a correlation matrix that can be inserted in the update loop of the network - the conceptor matrix. It will filter the state dynamics so that states that confess with the prinicipal directions of the described ellipses pass without modification while directions orthogonal to those will be suppressed.

2.2 Hierarchical Random Feature Conceptor

Here we present the hierarchical filtering and classification architecture proposed by Jaeger (2014). The changes that we made to it, so that it can accommodate the binocular rivalry condition, are highlighted in the following chapter. A schema of the complete architecture is shown in Figure **??**. The architecture consists of three identical copies of a random feature conceptor system, as it was introduced in the preceding chapter. They are arranged in a bi-directional hierarchy, making up three layers. At first we will introduce the state update equations that hold for each layer. These include mixing variables, which also dynamically evolve during the process of driving the system. Their impact on the mode of operation of the system will also be explained. In all equations the layer is denoted in the subscript by a place holder, l. In our current case of three layers, l can take values from the set $\{1, 2, 3\}$. The state update equations for all layers during the process of driving the system and collecting states are:

$$u_{[l]}(n+1) = (1 - \tau_{[l-1,l]}(n))y_{[l-1]}(n+1) + \tau_{[l-1,l]}(n)Dz_{[l]}(n) \tag{1}$$

$u_{[l]}$ is the input to a specific layer l. It is a mixture between two different constituents, the output of the lower layer $y_{[l-1]}$ and a self generated pattern of the current layer, $Dz_{[l]}$. The mixing variable $\tau_{[l-1,l]}$ defines the degree to which the input to the current layer is a self generated version of the pattern, and therefore carries some information about the certainty of the system that the current pattern it is producing is in fact the correct one. It is for this reason also referred to as a trust variable. For the case of $\tau_{[l-1,l]}$ being small, the current layer is driven by the lower layer to a larger extend, leaving the system in a more ”perceiving” than ”acting” mode. The trust variable $\tau_{[l-1,l]}$ and the associated driving modes of the system are discussed in more detail below. There exists one special case of $\tau_{[0,1]} = 0$, so that the sole input to the lowest layer of the hierarchy is just the signal $y_0(n)$.

$$r_{[l]}(n+1) = tanh(Gz_{[l]}(n) + W_{in}u_{[l]}(n+1) + b) \tag{2}$$

The reservoir states $r_{[l]}$ are comprised of information from the feature space activations $z_{[l]}$, as well as the input signal $u_{[l]}$ and the bias b. In particular, the feature space activations are mapped onto $r_{[l]}$ by the mapping G, and the input signal is fed into r through the input weights. The sum of all constituents is squashed by a $tanh$ nonlinearity, yielding the updated reservoir activations $r_{[l]}$.

$$z_{[l]}(n+1) = c_{[l]}(n). * Fr_{[l]}(n+1) \tag{3}$$

The feature space activations $z_{[l]}$ are computed by component-wise multiplication of the conception weights $c_{[l]}$ with the effect of the reservoir states $r_{[l]}$ in the feature space. In order to manoeuvre the reservoir activations into the feature state, they were projected by the mapping F. The adaptation of the conception weights $c_{[l]}$ is described further below.

$$y_{[l]}(n+1) = W_{out} r_{[l]}(n+1), \tag{4}$$

The output of layer l, $y_{[l]}$, is computed by application of the output weight vector W_{out} on the reservoir states $r_{[l]}$.

So far we have discussed the dynamic variables that change on a fast timescale, for all layers of the hierarchy. But there are more constituents to the whole system. The mappings D, G, F, W_{in} and W_{out} are all identical across the different layers of the hierarchy. G and F are initially sampled at random from a normal distribution. They resemble the functionality of the weight matrix in a usual reservoir computing setup. G is rescaled after a run of the system with white noise as input, to set the spectral radius of the sequential application of G and F. F remains unchanged. This procedure is further detailed in Jaeger (2014), Section 4.7. W_{in} is likewise sampled at random and also remains unchanged. W_{out} and D are learned after presentation of the prototype patterns, as it was presented in the previous Section **??**.

As the architecture is organised in a bi-directional hierarchy, there exists a bottom-up as well as a top-down flow of information. The output of the lower layer is fed to the higher layer, constituting the bottom-up flow. The top-down flow is the influence of conception weights of a higher module on a lower one. Both are mediated by the trust variables $\tau_{l-1,l}$. We will detail the both pathways a bit further.

The **top-down pathway** influences the conception weights in each layer of the hierarchy. The topmost layer is hereby a special case, as there is no layer above which can have any influence on its conceptor. By design its conceptor is constrained to be an aperture adapted disjunction of all prototype conceptors.

$$c_{[3]}(n) = \bigvee_{j=1,2} \varphi(c^j, \gamma^j(n)) \tag{5}$$

In the case of our binocular rivalry setting, these were the conceptors for two sinewaves of different periods that the system learned at training time, before the actual binocular rivalry simulation. $\gamma^j(n)$ are the aperture adaptation factors and $\varphi(c^j, \gamma^j(n))$ denotes the adaptation of the aperture of the conceptor c^j. More details on the disjunction of conceptors and on logical operations with conceptors in general can be found in Jaeger (2014).

With the constrained top-level conceptor one introduces a qualitative bias to the system, leading to the tendency to recognize something familiar in an input pattern and to generate a clean version of that familiar pattern. This means the system is de-noising its input, while noise is defined as everything in the signal that is not part of the recognised, familiar pattern. Adapting the top level conceptor $c_{[3]}$ is done by adapting the aperture of its constituent conceptors. As these aperture adapation factors are the mixing coefficients in the disjunction of prototype conceptors, they reflect the hypothesis of the system about the origin of the current driving signal. This hypothesis is passed downwards in the hierarchy. In each of the lower levels 2 and 1 an autoconceptor adaptation process is taking place, yielding module internal conception weight vectors $c_{[l-aut]}$. These are linearly mixed with the conception weight vector from the next higher layer, using the trust variables $\tau_{[l,l-1]}$ as mixing parameters.

$$c_{[l]}(n) = (1 - \tau_{[l,l+1]}(n))c_{l-aut}(n) + \tau_{[l,l+1]}(n)c_{[l+1]}(n) \tag{6}$$

If the trust variable is close to 1, the conception weight vector of the current level is largely determined by the next higher layer. In the other extreme case, where the trust variable is close to 0, the conception weight vector of the current level is largely determined by the autoconceptor on this level, leaving the opportunity for the input to influence. If the trust is low, the system is examining the input and identifying previously learned pattern within. On the other hand, if the trust is high, the system generates the pattern that belongs to the current hypothesis and is not sensitive to the driving input. The trust variables therefore steer the system between and actively generating and a passively perceiving mode. Jaeger (2014)

The **bottom-up pathway** influences the input to the higher levels 2 and 3. These levels have a self generated input simulation signal $Dz_{[l]}(n)$. Additionally to this, they receive the output from the next lower layer. The mixture coefficients $\tau_{[l-1,l]}$ determine how much influence the bottom-up pathway has against the self generated input simulation signal. If the trust variable is close to one, the module will generate a clean version of a prototype pattern. This, however, can also be a wrong one - the system is ”hallucinating”. One the other hand, if the trust variables are close to zero, the input is largely determined by the output of the lower module, and therefore would be running in an entirely externally driven mode with no effect of cleaning or noise suppression.

Putting everything together, the trust variables with their influence on the bottom-up and the top-down pathway steer the operation mode of the system between an action and perception mode.

2.3 The binocualr rivarly condition in a hierarchy of conceptor controlled RNN

In order to simulate the binocular rivalry condition we utilized the hierarchical conceptor controlled recurrent neural network, as it was presented in the preceding chapter. The reservoir was chosen to consist of 100 neurons and the feature space of 700 neurons. Bias scaling was 0.2 and the input was scaled by 1.2. The maps F, G, W_{bias} and W_{in} were sampled at random. The maps F and G, which replace the function of the weight matrix W, are rescaled so that the spectral radius of the subsequent application of both equals 1.4. Afterwards the system is run for 5600 simulated timesteps with white noise as input, and the network response is collected. The mapping G is recomputed on this data using regularized linear regression. The normalised root mean squared deviation (NRMSD) between the application of the old, unregularized G and the new G was 0.0003. Jaeger (2014) found this procedure to be necessary for stability reasons when running a system of random feature conceptors. More detail on this procedure can be found therein. Afterwards the system is run with the clean signals of two sinewaves in turn, in order to learn the prototype conceptors for them. Later, the binocular rivalry signal will be composed out of a superposition of these signals and noise. For every input pattern, the system is run through three periods:

For a washout period of 200 timesteps, during which the networks response starts to be correlated with the driver, no network responses are collected. Then the system is run in the conceptor adaptation mode for

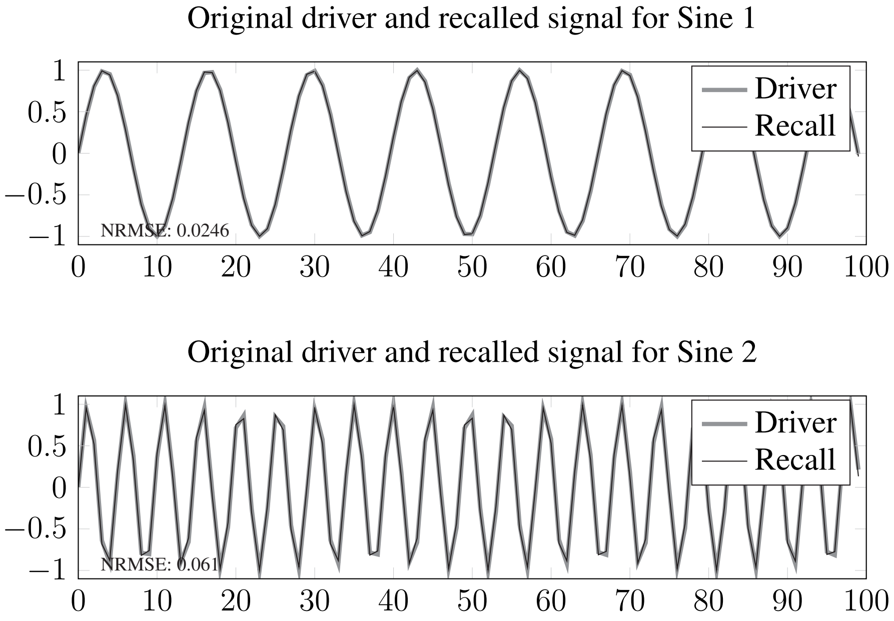

Figure 1. Successful loading of both sine patterns into the random feature conceptor architecture. The NRMSE between the original driver and the phase shifted, retrieved version of the pattern is on the order of 10^{-2} for both patterns. There is almost distinction visible to the eye between driver and recall.

2000 timesteps, wherein the prototype conceptor for that pattern is learned. Finally the system is run for 600 timesteps with the adapted conceptor in the network state update loop, and the network's response is collected. Again, all these three steps are gone through for every prototype pattern, in our case two sinewaves with different periods.

In the following, two learning steps are performed. The output weights W_{out} are computed by ridge regression with all collected reservoir states as arguments and the corresponding prototype patterns as targets. The NRMSD between the output of the system through the freshly calculated output weights and the prototype pattern is computed. The NRMSD in this case was 0.0027. In the second learning step, the loading, an input simulation matrix D is obtained. This is done by regularized linear regression, with the objective to reproduce the same network activations as they were elicited by the driver, but in absence of the driver. It serves the purpose of 'simulating the input', hence the name input simulation matrix. The NRMSD per neuron between the input driven network response and the network response elicited by D was 0.0005 on average per neuron.

Subsequently the success of the learning steps was tested by a recall period. For every pattern the trained system was run under the respective conceptor for 200 washout steps. This allows for the adaptation of the network dynamics to the control of the current conceptor. Afterwards the output of the system was collected for 200 timesteps and compared to the original prototype pattern. After the correction of an inevitable phaseshift, the NRMSD for the first sinewave with period 14.19 was 0.0246 and the NRMSD for the second sinewave with period 4.83 was 0.061.

So far we have seen the setup of one module of the random feature conceptor architecture with two sinewaves of different periods learned. In order to build a hierarchical random feature conceptor system, we bi-directionally connect three copies of this architecture. The resulting top-down and bottom-up pathways operate as it was introduced in the preceding chapter.

In addition to this we introduced a feedback loop from the top level hypothesis to the input of the system. This feedback loop suppresses those parts of the input signal that can be explained or predicted under the current hypothesis of the system.

Let us first introduce the composition of the input pattern, without the effect of the feedback loop.

2.3.0.1 Stimuli for binocular rivalry condition

The basis for the input to the system are the following two sinewaves that were mentioned earlier, with different periods, sampled at integer n:

$$s_1(n) = sin(\frac{2\pi n}{13.1900453}) \tag{7}$$

$$s_2(n) = sin(\frac{2\pi n}{4.8342522}) \tag{8}$$

Furthermore a component of normally distributed noise, with the signal to noise ratio of 0.5 with respect to the clean sinewave input, is added. The noise was found to be necessary to push the system into an oscillating regime. Figure 2 shows the components and the resulting input pattern.

The effective input signal to the system is different most of the time, due to the effect of the feedback loop. When the system settles on a hypothesis, the part of the input signal that can be explained by that hypothesis is subtracted from the input. Importantly, we defined the winning hypothesis by the procedure of 'the winner takes it all'. Therefore there is a winning hypothesis at all points in time, except for the cases where 0.5 is assigned as probability to both hypotheses. If one hypothesis is only slightly more likely, for example 0.55, while the other is assigned 0.45, the first one is decided to be the leading hypothesis. This effects the input drastically, the complete clear signal that belongs to the winning hypothesis is subtracted. Thereby the effective input to the system is usually a composition of noise and one signal source.

A sample of this effective input is shown in Figure 3.

The system is run for 50.000 timesteps. Over the course of this simulation the hypothesis of the system about the source of the driver is collected on all three levels of the hierarchy. Moreover the dynamics of the trust variables that operate between the levels are saved. The results and analysis of this simulation are detailed in the next section.

3 RESULTS

Figure 4 shows the results of the simulation for the first 3000 out of the total of 50000 simulated timesteps. The three topmost plots show the evolution of the hypothesis vectors for the three levels of the hierarchy. A few observations can be made: On level 1 the hypothesis are not yet really differentiated, there are relatively long periods where both hypotheses are almost equally likely. On level 2 this differentiation is far better, surpassed only by a little in layer 3. Moreover, a small delay in the processing of the system can be observed. Comparing level 2 and level 3 hypotheses, it can be seen that level 3 reacts similar but has a delay on the order of 100 to 300 timesteps with regard to level 2. Between level 1 and level 2 this is less obvious, but can also be observed. It is also far more difficult to see, because on level 1 the structure of the hypothesis peaks is still very different compared to the higher levels. Most importantly an oscillation between the hypotheses can be observed on all levels. The top level hypothesis vector can be interpreted as the perception of the system, switching from one sinewave to the other, back to the first one, and so

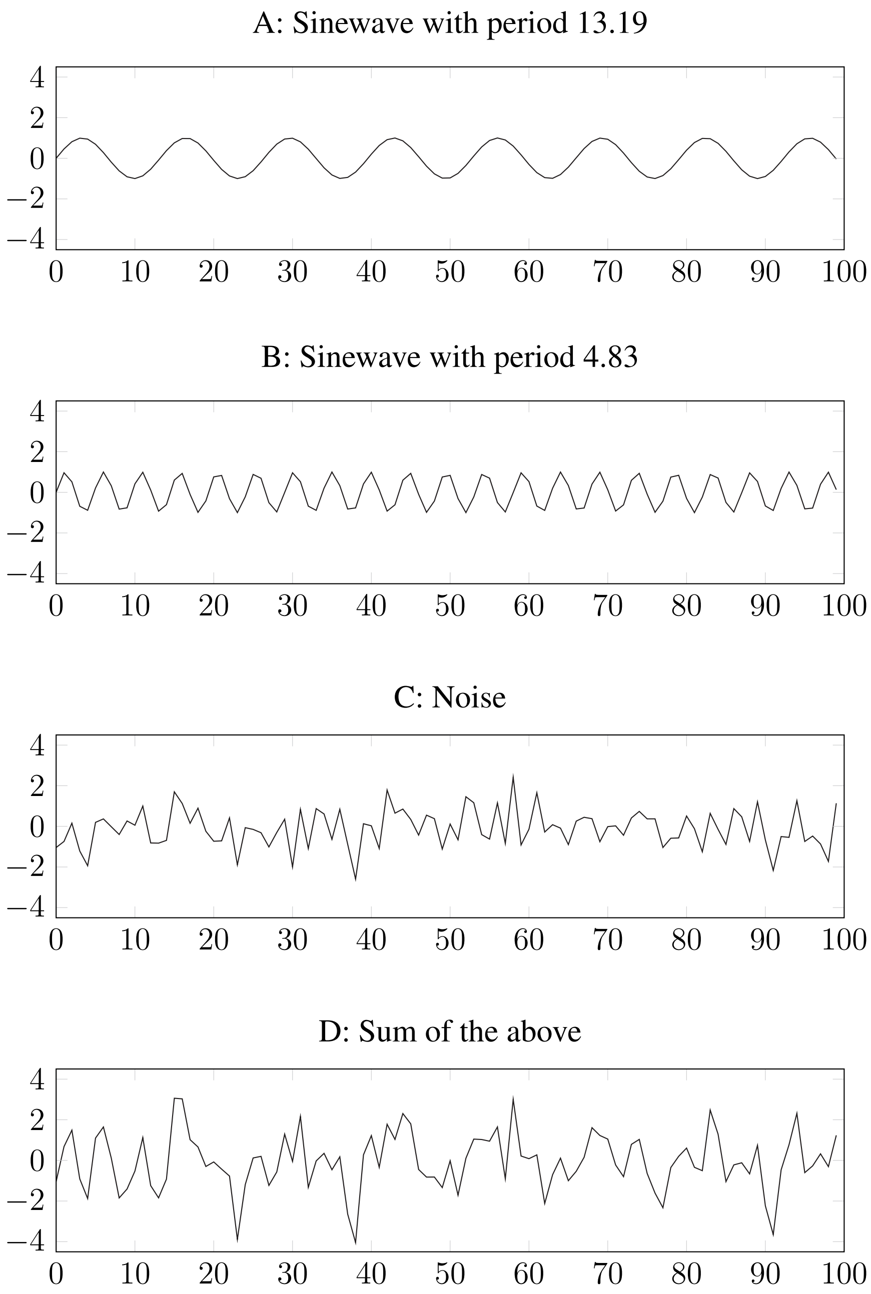

Figure 2. A sample of the binocular rivalry input, as it would be without the influence from the feedback loop. A and B show the familiar two sinewave patterns. C shows a sample of normally distributed noise. D shows the composed binocular rivalry input signal, effectively the sum of A, B and C

.

on. This resembles the perception human observers have when they are viewing rivalling stimuli. The bottommost plot displays the trust variables that operate between the levels. They both stay at a high level during the stimulation, indicating that the system is confident to generate the correct pattern most of the time. Especially for the trust variable between level 1 and 2 several small dips can be observed. These can correspond to a switch in input signal due to a change in hypothesis on the top level. The system realises that its prediction does not match the input pattern as much as it would, if it were to change its hypothesis and conceptor. It therefore operates shortly in an input driven manner to find the optimal input matching hypothesis and settles again, only to be tempted to change again as soon as the new hypothesis affects its input.

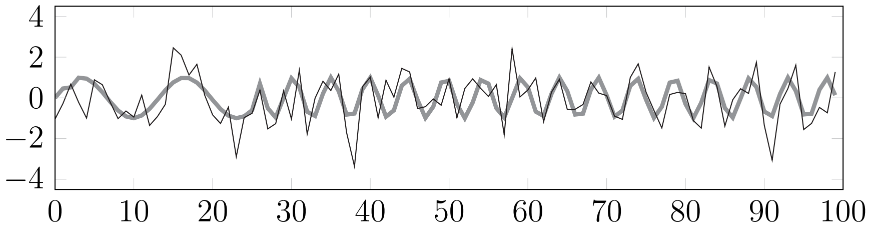

Figure 3. A sample of the effective binocular rivalry input, with influence from the feedback loop. Up to timepoint 23 the signal consists of sinewave 1 and noise, thereafter of sinewave 2 plus noise. The hypothesis that sinewave 2 is the source in the signal was winning until timestep 23. Therefore the signal of sinewave 1, which is not predictable under this hypothesis, remained in the input signal. From timestep 23 on the same reasoning holds, with hypothesis 1 being the winning hypothesis and sinewave 2 remaining as unpredicted residuum in the input signal.

Figure 5 shows the output of the top level module around transition points of hypotheses. This signal is the systems prediction of the input pattern. The plot shows how the predictions changes under the change of the leading hypothesis. Closely around the transition point the produced signal are slightly unstable, reflecting that the hypotheses are equally likely at these points in time. At the transition points the conceptor that controls the system is the disjunction of both patterns and the resulting prediction a combination of both patterns. One can observe this fact for example in the topmost plot between timestep 460 and 475. The peak has the overall width of the fist sinewave, but at the same time two more peaks of the period of the second sinewave ’on top’.

We calculated the distribution of dominance times on the data of the third level hypothesis vector. We in particular calculated the dominance times for each sinewave separately, in order not to get a mixed distribution that is skewed to either side because the patterns have a different signal strength. Both distributions are plotted in Figure 6. The distributions show similarity to the results of Levelt (1965). As Levelt did, we fit gamma functions to the data. For the general form

$$\frac{\beta^{\alpha}}{\Gamma(\alpha)}x^{\alpha-1}e^{-\beta x} \tag{9}$$

We estimated the parameters α and β using the `scipy.gamma` package for Python:

349

	α	β
sine 1	4.77	0.01684
sine 2	7.15	0.03982

These yield the following equation of the fit for sine 1

$$\frac{0.01684^{4.77}}{\Gamma(4.77)}x^{4.77-1}e^{-0.01684x} \tag{10}$$

and sine 2 respectively

$$\frac{0.03982^{7.15}}{\Gamma(7.15)}x^{7.15-1}e^{-0.03982x} \tag{11}$$

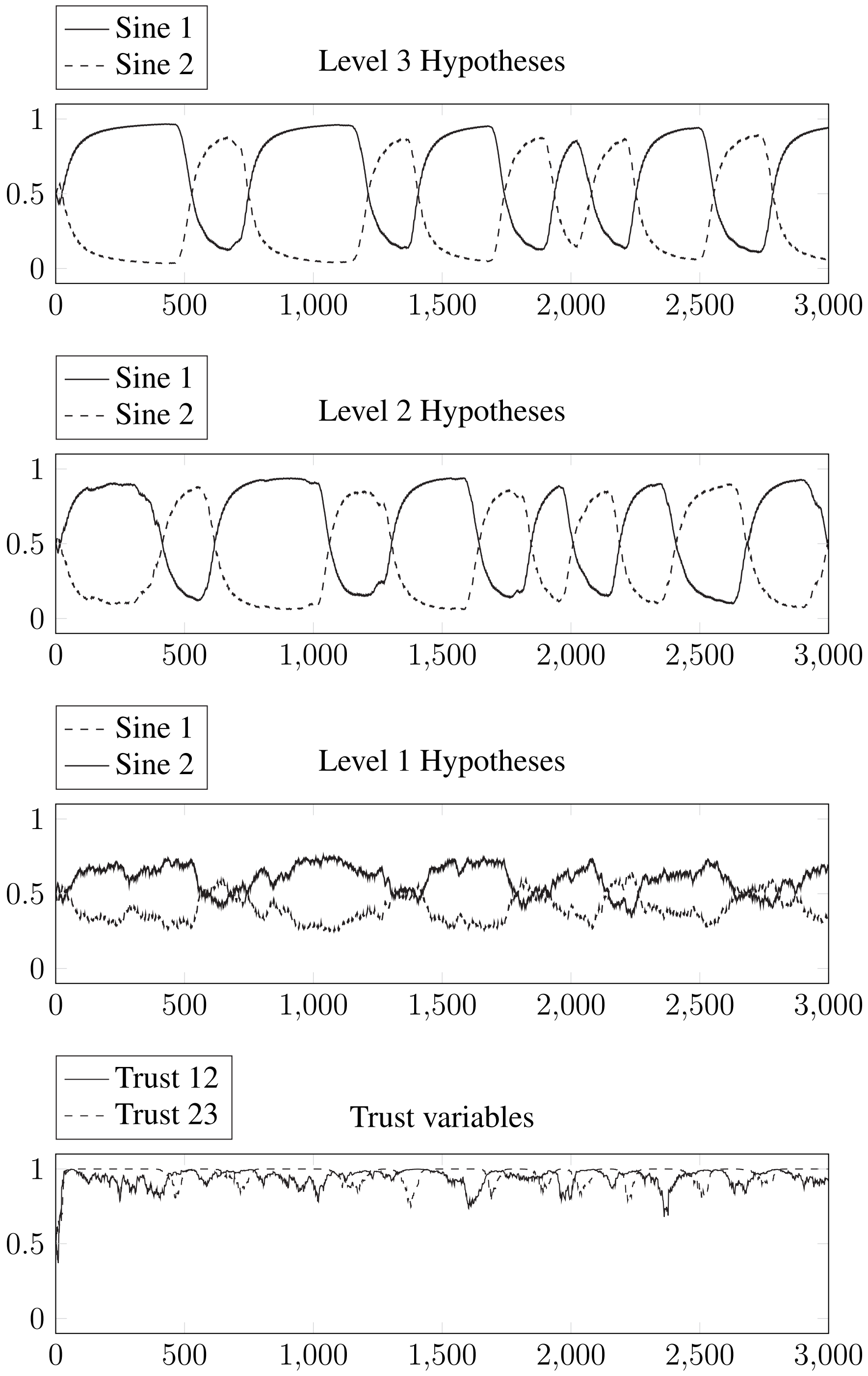

Figure 4. Results of the binocular rivalry simulation. Displayed are the first 3000 simulated timesteps. The three topmost plots show the hypothesis vectors for the three levels of the hierarchy. The bottommost plot shows the trust variables operating at the intersection of the levels of the hierarchy. For details see text.

$\Gamma(t)$ refers to the gamma function.

$$\Gamma(t) = \int_0^\infty x^{t-1} e^{-x}\, dx. \tag{12}$$

It extends factorials to all complex numbers except negative integers.

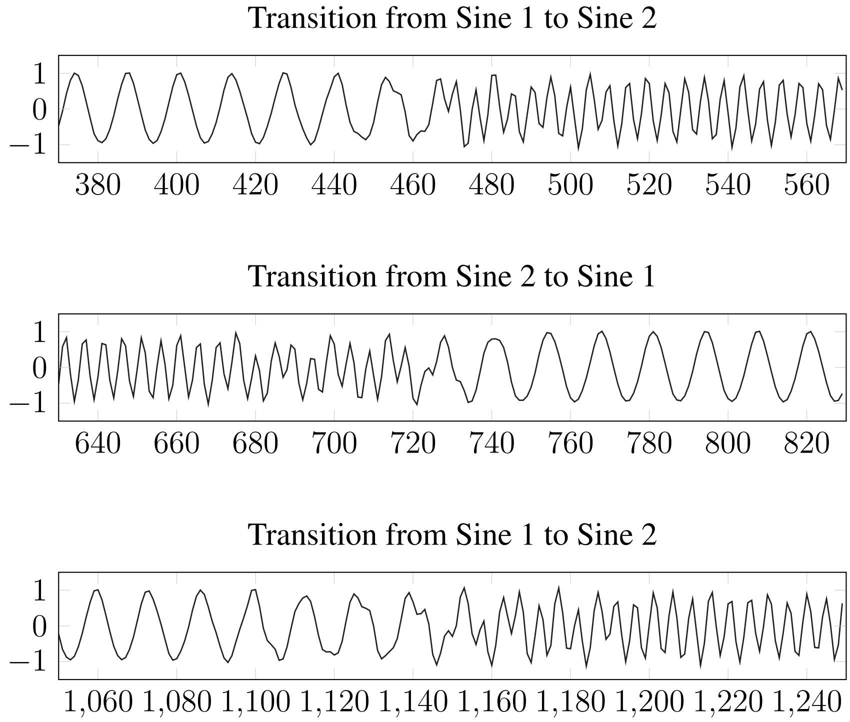

Figure 5. Output around transition points of hypotheses of the hierarchical random feature conceptors top level in the binocular rivalry condition. The switch in the generated signal can be observed clearly, as well as the production of a combination of both prototype patterns at the immediate point of transition.

4 DISCUSSION

We used a recent artificial neural network model, based on the paradigm of reservoir computing, to instantiate a simulation of the binocular rivalry in terms of predictive coding. This was a promising endeavour to us, since besides long standing efforts in the area of binocular rivalry, many open questions remain. In particular, we hope that we were able to deliver a first step into the direction of a model that is based on a general framework for perception.

The need for such a model was recently brought up from within the binocular rivarly research community in Hohwy et al. (2008). We emphasise however that our model is barely a first step into a promising direction. Even more so, we clearly point out that we can not make any full blown claims about the architecture actually working according to the ideas in the predictive coding framework. In the following we will detail our doubts. We have the intuition that coming short of real results in these points is not due to the fact that the direction of research leads nowhere. Quite the opposite, we think that waterproof results are coming up further along the way. We realise that we have only made the first steps in the direction of using the conceptor architecture for cognitive modelling tasks and we are sure that there is a lot to learn about these systems. Especially as the field of related mathematical research is very young, our study will greatly profit from upcoming results in that area.

4.1 Bayesian perceptual inference

In how far does our model perform Bayesian perceptual inference? Under the predictive coding theory the brain generates and tests hypothesis about the causes of the sensory data it encounters. In our simulation the system has learned two prototype patterns. These two patterns are ”the world” for the system. Besides the driving input itself, its internal representation of the prototype patterns is the only information it has

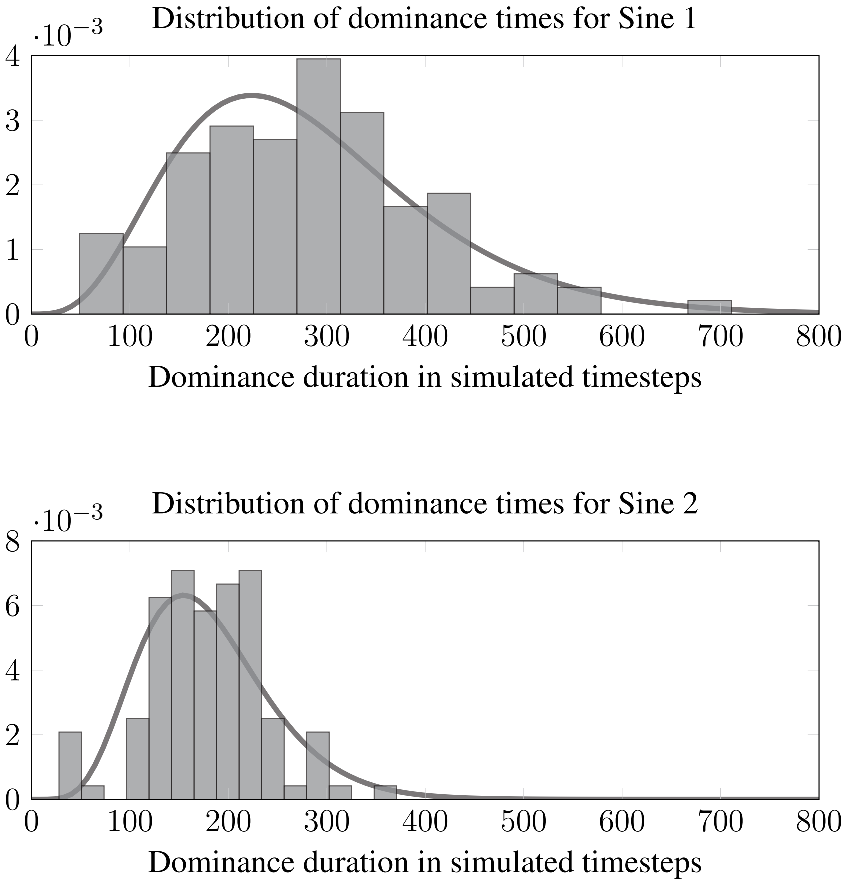

Figure 6. Distrubution of dominance times, seperately for each sinewave. Both histograms were fit to a gamma distribution function. The distribution of dominance times in the binocular rivalry simulation is similar to data acquired from experiments in humans, when they were viewing rivalling stimuli.

access to during runtime. As the system is also not adapting or learning any new patterns during the course of the binocular rivalry simulation, the only hypothesis it can make up involve the two prototype patterns. The simulation of the binocular rivalry condition shows that the system adapts its hypothesis about the current input in accordance to the input. On the level of the hypotheses it shows an alternating behaviour, just as it is the key observation in the binocular rivalry condition in humans. Insofar we have a working example of a challenging situation for a perceptual system. This of course is not sufficient to proof that the system is actually doing Bayesian perceptual inference. We believe a valid starting point to investigate whether a system is instantiating an inference mechanism of the kind that is inherent in humans, is to work with a system that has *no* clear issues that suggest it is *not* employing such a mechanism. In order to rigorously identify dynamics of the system with parts of the Bayesian theorem more work in that direction needs to be done. We had to leave this very interesting point to future investigations.

4.2 Low prior for compound hypothesis

As in the preceding paragraph discussed, the system has only the option to make up hypotheses from the two prototype pattern it knows. It can, however, settle on a mixture of these, maintaining for example the hypothesis that a mixture of the prototype patterns causes the current sensory input. This is in fact the case, if the system is run without the effect of the feedback loop. In many situations this is highly desirable and it is a research project in its own right in how far conceptor combinations really are able to combine concepts. Nevertheless in the special case of humans viewing binocular rivalry stimuli, the hypothesis of a mixture of both stimuli is a priori highly unlikely. Face-house compounds for example do usually not appear in the world. We tried to reflect this low prior probability for the compound hypothesis within a conceptor, but we were not able to construct it. This was mostly due to the notion of the negation of a conceptor. A conceptor is representing an ellipsoid in the networks state space, and its negation is defined by Jaeger (2014) as and ellipsoid spanned by the orthogonal directions of the original conceptor. A negative conceptor is not the complete state space which is not occupied by the original conceptor, as one might be tempted to think. This reasoning let us believe that we would not be able to build the low prior for the compound hypothesis into a conceptor. In the end we circumvented this issue by the construction of the feedback loop. The signal corresponding to the winning[1] hypothesis is completely subtracted from the input signal. Therefore the actual input to the system always corresponds only to one signal plus the added noise. This design choice can be supported by the argument of a strong effect of the prediction of the system on the actual perception. The reasoning is that the predicted signal is completely explained and therefore can be subtracted from the input signal. We employed this mechanisms, however, it was not the original approach. In the original approach we tried to take the bare prediction of the system on the top layer and subtract that from the input. This turned out to be not suitable for our attempt, as reservoir systems as we use them produce inevitable phase shifts of the generated signal versus the input signal. Moreover we were stuck with the just mentioned problem of the system believing that the current input is a mix of both signals. We therefore construct the input signal on the basis of the hypothesis vector only, and not with the influence of direct feedback from the top layer prediction of the system.

4.3 Prediction error minimization

The hierarchical random feature architecture tries to minimize prediction error by selecting the best hypothesis in order to predict the incoming sensory data. The residuum of the incoming data which can not be explained is called the prediction error. In contrast to predictive coding the proposed architecture does not signal the prediction error upwards in the hierarchy, but a denoised version of the sensory input. 'Denoised' means in this context that parts of the signal which are not predictable under the current hypothesis are regarded as noise and are suppressed. This in fact leads to less prediction error on higher layers of the hierarchy, as the prediction error is suppressed by each layer. This mechanism is therefore actually minimizing prediction error, but in a slightly different fashion than the usually in predictive coding proposed upwards signalling of the residual signals or prediction errors. Minimizing prediction error just by suppressing all signals that can not be predicted on its own does not seem very useful. But this process is aided by a general assessment of fit of all prototype patterns to the input signal. This is inherent in the conceptor mechanism. Therefore the mechanism for prediction error minimization is different in the hierarchical random feature conceptor as compared to the usual notion in predictive coding. This issue is still in debate, also for the predictive coding research community, as we are not aware of any clear cut evidence in favour of and against other possible realisations of error signalling.

4.4 Comparing dominance times to Levelt's work

The distribution of dominance times that we obtained from the simulation is of the same form as the dominance time distribution of Levelt's work in the 60s. We are not sure why our architecture changes its perception with these statistical regularities. The shape of the dominance times histogram might even be due to the nature of the noise that is added to the stimulus. The similarity, by itself, is a success for us. We set out to this endeavour without knowing if we would find encouraging results on the way. It makes us

[1] The winning hypothesis, as explained in the preceding section, is the hypothesis that is assigned the larger probability, even if it is only in slight advance.

curious that the distribution of dominance times is so similar between our, not parametrically optimised setup, and data from the real world. Still, we can not draw any claims from the observation at this point. Rather, we are at a point where plenty of directions of future research are on offer.

DISCLOSURE/CONFLICT-OF-INTEREST STATEMENT

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

AUTHOR CONTRIBUTIONS

The statement about the authors and contributors can be up to several sentences long, describing the tasks of individual authors referred to by their initials and should be included at the end of the manuscript before the References section.

ACKNOWLEDGMENTS

...

Funding: ...

SUPPLEMENTAL DATA

- Introduction: Succinct, with no subheadings.
- Materials and Methods: This section may be divided by subheadings. This section should contain sufficient detail so that when read in conjunction with cited references, all procedures can be repeated.
- Results: This section may be divided by subheadings. Footnotes should not be used and have to be transferred into the main text.
- Discussion: This section may be divided by subheadings. Discussions should cover the key findings of the study: discuss any prior art related to the subject so to place the novelty of the discovery in the appropriate context; discuss the potential short-comings and limitations on their interpretations; discuss their integration into the current understanding of the problem and how this advances the current views; speculate on the future direction of the research and freely postulate theories that could be tested in the future.

Supplementary Material should be uploaded separately on submission, if there are Supplementary Figures, please include the caption in the same file as the figure. LaTeX Supplementary Material templates can be found in the Frontiers LaTeX folder

Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text.

REFERENCES

Clark, A. (2013). Whatever next? predictive brains, situated agents, and the future of cognitive science. *Behavioral and Brain Sciences* 36, 181–204

Gähde, U., Hartmann, S., and Wolf, J. H. (2014). *Models, Simulations, and the Reduction of Complexity*, vol. 4 (Walter de Gruyter)

Hohwy, J. (2013). *The predictive mind* (OUP Oxford)

Hohwy, J., Roepstorff, A., and Friston, K. (2008). Predictive coding explains binocular rivalry: An epistemological review. *Cognition* 108, 687–701

Jaeger, H. (2001). The ”echo state” approach to analysing and training recurrent neural networks-with an erratum note. *Bonn, Germany: German National Research Center for Information Technology GMD Technical Report* 148, 34

Jaeger, H. (2014). Controlling recurrent neural networks by conceptors. *arXiv preprint arXiv:1403.3369*

Levelt, W. J. (1965). *On binocular rivalry*. Ph.D. thesis, Van Gorcum Assen

FIGURES

Figure 7. Enter the caption for your figure here. Repeat as necessary for each of your figures