

Article

Not peer-reviewed version

---

# It Is What It Isn't: Introducing a Constraint-Based Approach To Structure Learning

---

[Christoffer Lundbak Olesen](#) , Nace Mikuš , [Mads Hansen](#) , Nicolas Legrand , [Peter Thestrup Waade](#) \* ,  
Christoph Mathys

Posted Date: 9 February 2026

doi: 10.20944/preprints202602.0668.v1

Keywords: structure learning; constraint-based dynamics; cognitive modelling; delusions; hierarchical Gaussian filter; computational psychiatry



Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a [Creative Commons CC BY 4.0 license](#), which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Disclaimer/Publisher's Note: The statements, opinions, and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions, or products referred to in the content.

Article

# It Is What It Isn't: Introducing a Constraint-Based Approach To Structure Learning

Christoffer Lundbak Olesen <sup>1</sup>, Nace Mikuš <sup>1</sup>, Mads Hansen <sup>2</sup>, Nicolas Legrand <sup>1</sup>, Peter Thestrup Waade <sup>3,\*</sup> and Christoph Mathys <sup>1</sup>

<sup>1</sup> Interacting Minds Centre, Aarhus University, Jens Chr. Skous Vej 4, 8000 Aarhus C, Denmark

<sup>2</sup> Department of Philosophy, University of Bristol, Cotham House, Bristol BS6 6JL, United Kingdom

<sup>3</sup> Translational Neurmodelling Unit, University of Zurich and ETH Zurich, Wilfriedstrasse 6, 8032 Zurich, Switzerland

\* Correspondence: pthestrup@ethz.ch

## Abstract

Biological cognition depends on learning structured representations in ambiguous environments. Computational models of structure learning typically overlook the temporally extended dynamics that shape learning trajectories under such ambiguity. In this paper, we reframe structure learning as an emergent consequence of constraint-based dynamics. Informed by a literature on the role of constraints in complex biological systems, we build a framework for modelling constraint-based dynamics and provide a proof-of-concept computational cognitive model. The model consists of an ensemble of components, each comprising an individual learning process, whose internal updates are locally constrained by both external observations and system-level relational constraints. This is formalised using Bayesian probability as a description of constraint satisfaction. Representational structure is not encoded directly in the model equations but emerges over time through the interaction, stabilisation, and elimination of components under these constraints. Through a series of simulations in environments with varying degrees of ambiguity, we demonstrate that the model reliably differentiates the observation space into stable representational categories. We further analyse how global parameters controlling internal constraint and initial component precision shape learning trajectories and long-term behavioural alignment with the environment. The results suggest that constraint-based dynamics offer a viable and conceptually distinct foundation for modelling structure learning in adaptive systems. We further analyse how global parameters controlling internal constraint and initial component precision shape learning trajectories and long-term behavioural alignment with the environment. We show that this allows to capture structure learning even in cases where it is maladaptive, such as delusion-like belief updating.

**Keywords:** structure learning; constraint-based dynamics; cognitive modelling; delusions; hierarchical Gaussian filter; computational psychiatry

## 1. Introduction

Biological cognitive agents need to learn from undifferentiated observations, often while facing an ambiguous environment. To function effectively under such conditions, they must differentiate these observations to support adaptive behaviour. This requires the construction of an internal representational structure that determines what differentiable categories perceptually exist for the organism. This capacity to identify and organise representational categories is here referred to as structure learning [1].

In the modelling literature, structure learning is typically framed as a process of selecting or inferring an underlying structure of entities or categories [2]. For structure learning in Bayesian network analysis, the goal is to infer the most probable set of dependencies among a fixed set of variables [3]. Within the active causal learning paradigm, structure is inferred through intervention in a

system of covariant variables [4]. Such approaches focus on modelling a latent structure of pre-labelled data, i.e., relations between known categories. However, since the world does not present itself with labels, the structure learning of biological cognitive systems must include the more fundamental challenge of identifying the natural categories of the world.

Although recent computational cognitive modelling of structure learning spans different cognitive domains and varies in scope, many include the discovery of novel categories [5–9]. In these works, the term *category* might have a slightly different conceptual framing, often interpreted as either object class, abstraction, perceptual concept, or latent cause, but is, regardless of interpretation, modelled as a functional component comprising a set of parameters, which are added to or developed within a larger structure of components during learning. The unique parametrisation of each component describes the “thing” or concept in the environment that it perceptually represents, determining the likelihood of each observation belonging to the given component. The inferred structure then comprises a differentiation of the observation space into discrete categorical representations.

While some of these approaches support online learning, i.e. incremental learning from a stream of observations, they are typically not formulated to capture the temporally extended dynamics of structure formation and how these shape learning trajectories under environmental ambiguity, which we take to be key aspects of biological cognition. Here, we present a novel computational framework that explores structure learning as a temporal, dynamic process. In the next sections, we outline the conceptual and theoretical background for this approach and describe the framework before providing a general proof-of-principle that our framework is capable of modelling structure learning.

### 1.1. A Constraint-Based Approach to Structure Learning

Rather than treating structure learning as the selection among candidate representations, we model it as a process in which representational structure is continually shaped by constraint-based dynamics. The goal is not to develop a comprehensive cognitive architecture, but to demonstrate that internal differentiation of perceptual categories can emerge as a result of such dynamics.

This approach is informed by work within theoretical neuroscience [10–12], complex systems theory [13], theoretical biology [14–17], and philosophy [18–21] that emphasises the role of constraints in adaptive dynamical systems. Constraints come in many forms. Scaffolds, buffers, attractors, entrenchments, boundaries, initial conditions, and priors are all examples of constraints, since they limit some state, process, entity, or event. An often overlooked property of constraints is that they are relational (for a comprehensive review of the properties of constraints, see Hansen & Olesen, forthcoming). Something is a scaffold for the process it supports, a boundary for the domain it delimits, or a prior for the inference it informs. If one respectively takes away the process, domain, or inference in these examples, one leaves no scaffold, boundary, or prior. In short, it is by being in a certain relation that something comes to constrain something else.

Constraints are often thought of as mere limitations posing a restriction on a set of possibilities, i.e. fewer options, outcomes, effects, etc., are possible (or probable) in light of the constraint. Although limitation *is* a defining feature of constraints, their effects go beyond. Constraints can also enable new possibilities as a consequence of being limiting. As an intuitive example, games like chess are defined by strict limitations on how and when players can move their pieces. The possibility of the game itself and all its openings and strategies is enabled by these limitations. If we gave players the freedom to move their pieces anywhere at any time, there would be no game of chess. Similarly, traffic lights enable efficient traffic flow by selectively limiting the movement of cars, and our skeleton enables our capacity to walk by limiting the possible movement of our joints [13]. The commonality here is that by constraining elements at one level, these elements are stabilised into otherwise unlikely patterns or structures necessary for realising a phenomenon at a different level. As such, constraints play an important role for emergent capacities, and in biological systems such capacities include self-organisation, agency, and cognition. Our present purpose here is to model structure learning as an emergent capacity enabled by constraints on lower-level learning processes.

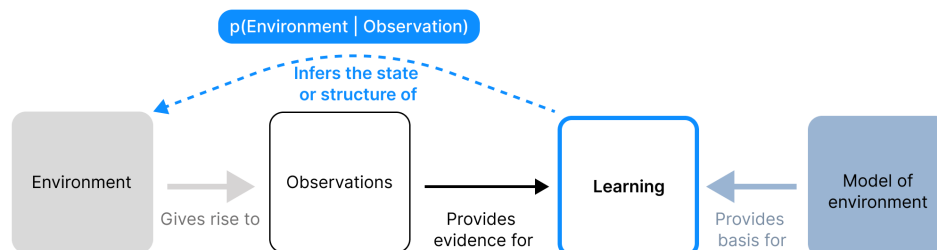
Since a constraint cannot be anything other than a relation, constraints should be modelled as relations. In probabilistic terms, such relations can be expressed in conditional probabilities, allowing for utilising a Bayesian formalism. However, we explicitly distinguish this from the more common notion of Bayesian inference. In Bayesian inference, Bayes' theorem is understood epistemically as a formal rule for optimally integrating new evidence with prior knowledge according to the rules of probability. When modelling cognitive agents as Bayesian inference agents, this rule is often understood as a temporal sequence starting with a prior that is evaluated against some evidence using the likelihood, then yielding a posterior. This maps a certain procedure for calculating a conditional probability onto a temporal sequence of an epistemic inference procedure that the cognitive agent is performing. However, the relationships between the quantities that Bayes' theorem consists of are not inherently temporal or sequential but logical. Here, we use this Bayesian logic as a framework for probabilistically expressing how constraints are satisfied while system dynamics unfold. This conceptually reframes the elements of Bayes' theorem as non-sequential and non-epistemic, i.e. as quantities whose relationships define the logical conditions under which a set of constraints is satisfied at a given moment in time. This line of thought has led us to the design of our current structure learning model, and while the resulting computational procedure does not wholly contradict a standard epistemic interpretation (i.e. as explicit inference), we provide this perspective throughout the paper to clarify the modelling assumptions, motivate the model design, and highlight how constraint-based interpretations may offer an alternative framing for modelling cognitive systems.

We distinguish between *component-level* and *system-level* perspectives, both conceptually and formally. The component-level describes individual learning processes in isolation, whereas the system-level describes the ensemble of components and their interactions. In this work, observations play different roles at the two levels. At the component-level, observations provide evidence for learning in the typical sense of Bayesian inference. At the system level, observations instead function as external constraints on the activity of each component. As a consequence, even though all learning processes at the component-level "see" all observations the same, they do not all learn equally from them. Learning is selectively modulated, such that for a given observation some components update strongly while others update weakly or not at all. Thus, categorisation is an emergent system-level property defined by differentiation within the ensemble. Because each component functions in isolation at the component-level with equal access to all observations, no individual component can by itself carve the observation space into distinct categories. This differentiation arises only at the system-level where the meaning of a component can be defined in opposition to other components: it is what it isn't.

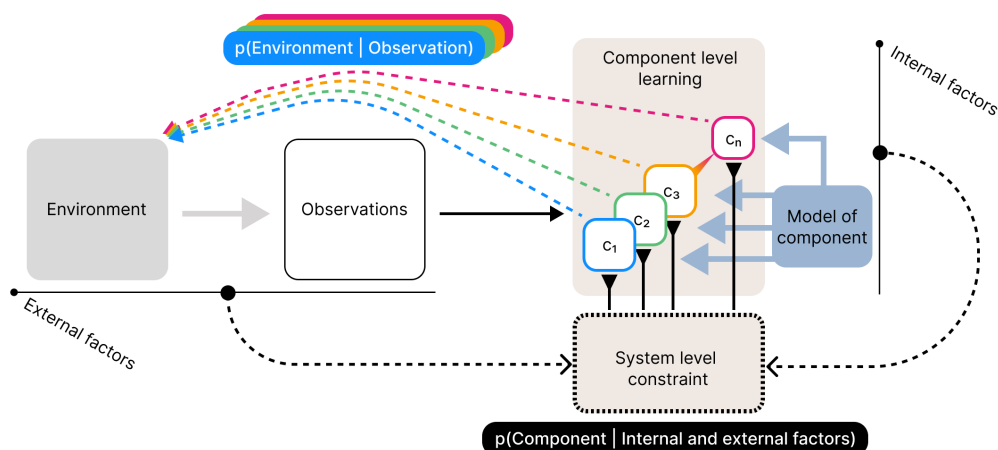
Importantly, this means that the capacity we aim to model, i.e., structure learning, is not directly present in the formal description of the model, as it would be in more standard Bayesian inference models (see Figure 1). The set of equations and the algorithmic procedure we present instead specify interaction dynamics and constraints, and structure learning is solely modelled through the unfolding of these dynamics over time, i.e., by forward simulation.

The model we present here treats structure learning as the result of ongoing interactions between internal and external constraints. Internal constraints capture the limitations imposed on the system dynamics from within, thus referring to the system's internal context, while external constraints are limitations imposed by the system's external context. In the present implementation, the external constraint is given by the current observation, but more generally external constraints may reflect any factor in the external context. Likewise, internal constraints may in principle be operationalised differently from how we do so here without violating the general framework. The central premise is that, through the interaction of these internal and external constraints, the system continually reshapes itself through local dynamics within its developing structure, rather than selecting among discrete representational alternatives. Here, a stable set of representational categories emerges as some representations becoming increasingly influential over time, while others fade.

### A) General scheme of Bayesian learning models



### B) Constraint-based structure learning



**Figure 1.** Conceptual schemes of learning models. **A)** The general scheme of Bayesian learning models. Here learning is about inferring the state (or structure in the case of structure learning) of the environment. This is done using observations as evidence for the inference scheme on the basis of a model of how the environment generates these observations. Thus, the model tries to mirror the generative process of the environment and for this reason is called a generative model. This general scheme also applies to Bayesian structure learning, where the generative model additionally includes the way structure is generated. Note, that while the inference is here expressed in the simplistic form as the probability of environment given observation, it is in principle always conditioned on the model as well. **B)** Scheme of our constraint-based structure learning model. Here, instead of having one learning process with a generative model of the environmental structure, we have a system of component learning processes, denoted  $c$ . Each component follows the general scheme of Bayesian learning, but instead of having a generative model of the environment as a whole, it has a model of environmental components, i.e. a model of how an individual "thing" generates observations. All components receive all observations but how much they learn from them is constrained from the system-level by both internal and external factors. Here the constraint on the components is represented by the probability of the component given these internal and external factors. As this is not understood in terms of inference we read it as an expression of how much the learning activity of the components are suppressed by these constraints. The structure learning unfolds dynamically as the component learning processes are constrained into a representational structure of the environment.

This framework turns out to be particularly useful when considering structure learning in ambiguous environments that cannot be truly and fully differentiated, in principle. In the following section, we provide a formal specification of the model and describe the simulation environment, experiments, and evaluation measures used in this study.

## 2. Materials and Methods

The model consists of a set of components added incrementally over time. Each component implements an individual learning process that tracks the central tendency and variability of a stream of observations. We distinguish between a *component-level* description of the internal learning processes and a *system-level* description of the ensemble, which governs component activity and determines how components contribute to the evolving representational structure.

The model receives a stream of one-dimensional real-valued observations  $o_t \in \mathbb{R}$  over discrete timesteps  $t$ . Observations are generated by a set of environmental sources, each associated with a distinct distribution in the observation space. Each such distribution is referred to as an *environmental component*, i.e. a “thing” that gives rise to observations in a particular region of the observation space. At each timestep, a single observation is generated by a randomly selected environmental component, after which it becomes inaccessible. While the environment has a stable underlying structure, the model is exposed to it one observation at a time, and the generating source of any given observation is latent. Depending on the distance between environmental components, observations may therefore be more or less ambiguous due to overlap in observation space (see Figure 2A).

At each timestep, the model encounters an observation  $o_t$  and proceeds through the following update cycle:

1. A new component is instantiated and added to the model, anchored on the current observation.
2. At the system level, an activity level is computed for each component.
3. At the component level, established components update their learning state proportionally to their activity level (the newly added component is excluded from this step).
4. Components with insufficient weight are removed from the model.

The overall model structure is formed by repeating this cycle over time, allowing components to be added, updated, and removed throughout learning.

The model includes three system-level free parameters and the free parameters that define the component-level. For orientation, we summarise the roles of all parameters here, before specifying their precise effects below. At the system level,  $\alpha$  sets the threshold for component removal and determines a baseline propensity for structural expansion. This propensity is further modulated by  $\sigma$  and  $\lambda$ , which respectively control the initial precision of newly instantiated components’ representational capacity and the general level of internal constraint. To model the component-level learning process, various learning models can be used. Here, we use the Hierarchical Gaussian Filter (HGF), described by one free parameter  $\omega$ . Unless otherwise specified, we use the following standard parametrisation throughout the simulations:

$$\sigma = 0$$

$$\lambda = 0.7$$

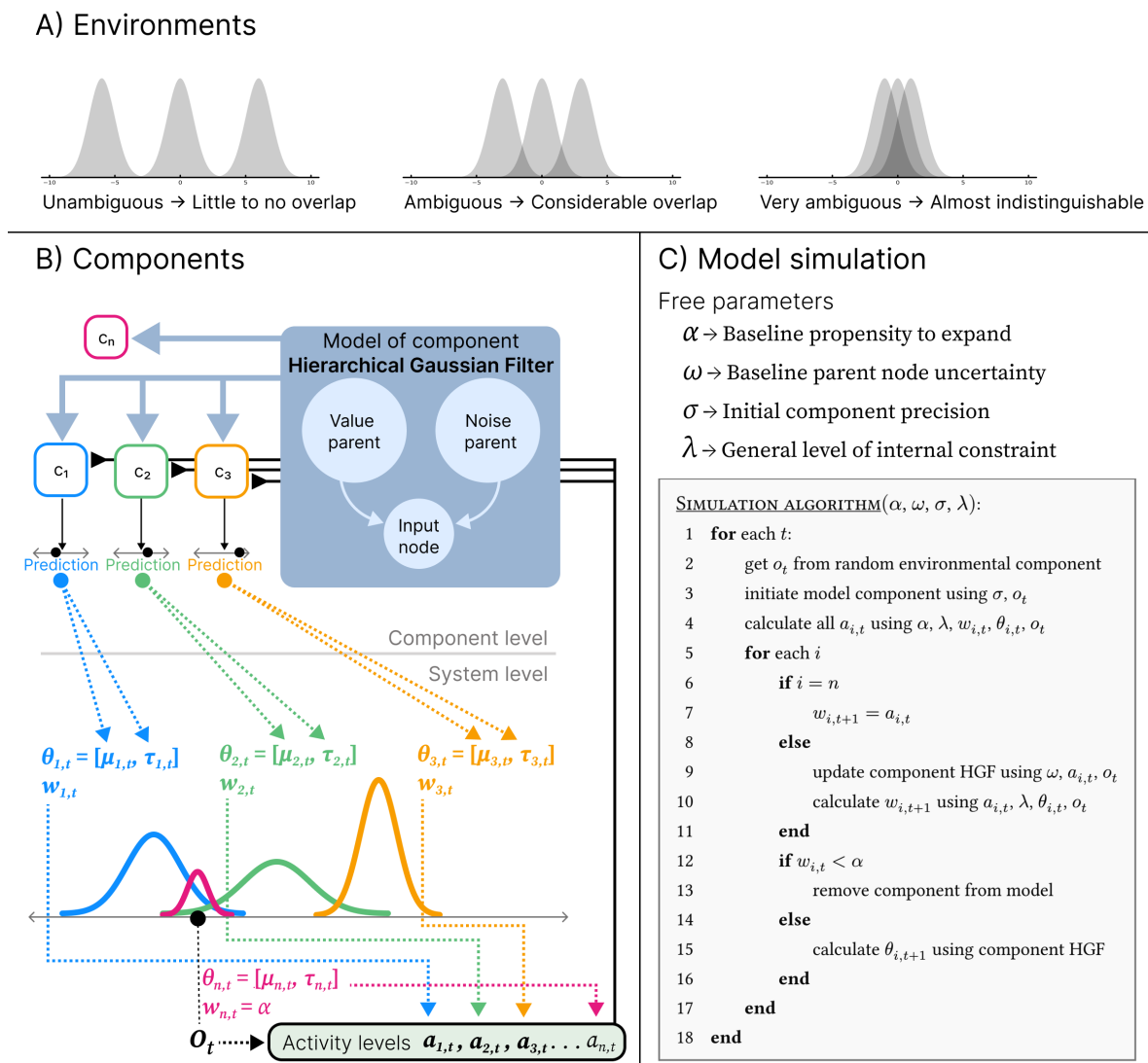
$$\alpha = 0.01$$

$$\omega = -6$$

These values were chosen based on exploratory tests and visual inspection aimed at identifying a region of the parameter space in which the model exhibits stable behaviour across conditions.

### 2.1. Component-Level

To describe the learning process at the component-level we use the HGF, a generalized modelling framework for inverting networks of hierarchically coupled random walks (refer to [22] for details).



**Figure 2.** Overview of the simulation environments and model procedure. **A)** Depiction of environments. Each Gaussian distribution correspond to a single environmental component. The depicted environments consists of three components of varying ambiguity, which is defined by the amount of overlap between environmental components. The three environments depicted correspond to the unambiguous, ambiguous and very ambiguous environments used in the simulation experiments. **B)** Schematic of model components and the relation between component-level learning and system-level organisation. At the component level, each component is an isolated instance of an individual learning process implemented here as a simple Hierarchical Gaussian Filter (HGF) serving as a generative model of an environmental component. Every timestep the HGFs produces a prediction about incoming observations. At the system level, each component is represented as a candidate category with a Gaussian representational capacity distribution over the observation space, parametrised by  $\theta_{i,t} = [\mu_{i,t}, \tau_{i,t}]$  and a weight  $w_{i,t}$ , where  $\theta_{i,t}$  is derived from the current HGF predictions (depicted by colored dashed lines traversing from component-level to system-level). At each timestep  $t$ , a new component  $c_n$  is instantiated with representational capacity centred on the observation  $o_t$ , representing a candidate expansion of the representational structure. A distribution of activity levels  $a_{i,t}$  across components, is calculated from the representational capacity of components and the observation (depicted by the arrows pointing to the activity levels box). These activity levels modulate learning by scaling the component-level update of each established HGF (depicted by the solid black lines with a reverse arrowhead; note that they continue behind the HGF box). **C)** Free parameters and pseudocode for the simulation algorithm.

Here, we use a simple HGF structure consisting of a single continuous input node with a value parent and a noise parent. The value parent tracks the central location of the observation in observation space, while the noise parent tracks the variability of the observations. At each timestep  $t$ , the input node of the  $i$ -th component makes a prediction about the mean and precision of the observations at the next timestep  $t + 1$ , based on the predictions of the parents:

$$\hat{\mu}_{i,t}^{\text{input}} = \hat{\mu}_{i,t}^{\text{value}} \quad (1)$$

$$\hat{\pi}_{i,t}^{\text{input}} = \frac{1}{\exp(\zeta_i^{\text{input}} + \mu_{i,t-1}^{\text{noise}})} \quad (2)$$

Here the superscript denotes what HGF node the variable belongs to and hat denotes that it is a prediction. Note that  $\zeta_i^{\text{input}}$  is a constant denoting the input node's tonic observation noise, and with Equation (2) controls its baseline precision prediction. Parent predictions are computed similarly for both parent types:

$$\hat{\mu}_{i,t}^{\text{parent}} = \mu_{i,t-1}^{\text{parent}} \quad (3)$$

$$\hat{\pi}_{i,t}^{\text{parent}} = \frac{1}{\frac{1}{\pi_{i,t-1}^{\text{parent}}} + e^\omega} \quad (4)$$

where  $\omega$  is an HGF parameter which controls the baseline uncertainty of the parent, which in turn controls the learning rate. This could in principle vary across nodes, but in our model we set  $\omega$  as a global fixed parameter for all nodes. The mean and precision of both parent nodes are updated at every timestep:

$$\pi_{i,t}^{\text{value}} = \hat{\pi}_{i,t}^{\text{value}} + \hat{\pi}_{i,t}^{\text{input}} a_{i,t} \quad (5)$$

$$\mu_{i,t}^{\text{value}} = \hat{\mu}_{i,t}^{\text{value}} + \frac{\hat{\pi}_{i,t}^{\text{input}}}{\pi_{i,t}^{\text{value}}} \delta_{i,t} a_{i,t} \quad (6)$$

$$\pi_{i,t}^{\text{noise}} = \hat{\pi}_{i,t}^{\text{noise}} + \frac{1 + \epsilon_{i,t}}{2} a_{i,t} \quad (7)$$

$$\mu_{i,t}^{\text{noise}} = \hat{\mu}_{i,t}^{\text{noise}} + \frac{\epsilon_{i,t}}{2\pi_{i,t}^{\text{noise}}} a_{i,t} \quad (8)$$

where  $\delta$  and  $\epsilon$  denotes value and noise prediction error respectively:

$$\delta_{i,t} = o_t - \hat{\mu}_{i,t}^{\text{input}} \quad (9)$$

$$\epsilon_{i,t} = \frac{\hat{\pi}_{i,t}^{\text{input}}}{\pi_{i,t}^{\text{value}}} + \hat{\pi}_{i,t}^{\text{input}} \delta_{i,t}^2 - 1 \quad (10)$$

The variable  $a_{i,t}$  denotes the activity level for the  $i$ th component at time  $t$  calculated at the system-level. This variable is not part of the HGF framework, but an addition to the update equations, which is central to our current modelling approach. Note that the form of some of the above equations, as well as the updating narrative, has been simplified in the light of our modelling context. For a full account, refer to [22]. However, besides adding the activity term  $a_{i,t}$  the above is mathematically equivalent with the HGF framework.

New components are initialised with the following values:

$$\zeta_n^{\text{input}} = \sigma \quad (11)$$

$$\mu_{n,t}^{\text{value}} = 0_t \quad (12)$$

$$\pi_{n,t}^{\text{value}} = 1 \quad (13)$$

$$\mu_{n,t}^{\text{noise}} = 0 \quad (14)$$

$$\pi_{n,t}^{\text{noise}} = 1 \quad (15)$$

where  $n$  denotes the number of components and consequently, as a subscript, also denotes the newest added component. Here  $\sigma$  is a global system-level parameter. Note that in the case that  $\sigma$  changes during the course of learning, it is possible for different components to have different values of  $\zeta_i^{\text{input}}$ , as they inherit the  $\sigma$  value at the time of their initialisation. However, while  $\zeta_i^{\text{input}}$  is constant, it is always used in summation with  $\mu_i^{\text{noise}}$  (Equation (2)), and can therefore be seen as setting an initial condition, i.e. a constraint that the learning process can overcome by adjusting  $\mu_i^{\text{noise}}$  relative to  $\zeta_i^{\text{input}}$ . Or in epistemic terms,  $\sigma$  functions as a prior precision for new components at the component-level, specified at the system-level.

## 2.2. System-Level

In the previous section we described the individual learning processes at the component-level. Now we turn to the system-level and start by describing how components are viewed at this level. We note that at this level, it is irrelevant for the general form of the model what learning processes are implemented at the component-level, although some specifics are tailored to work with the HGF in this instance of the model (e.g.  $\sigma$ ).

### 2.2.1. Components

Each model component functions as a candidate representational category, comprising a Gaussian distribution over the observation space. This distribution characterises the *representational capacity* of the component, i.e. the degree to which a component is able to represent an observation at a given location in the observation space. The  $i$ th component's representational capacity at time  $t$  consists of a Gaussian distribution parametrised by  $\theta_{i,t} = [\mu_{i,t}, \tau_{i,t}]$  and a weight  $w_{i,t}$ . Here  $\theta_{i,t}$  is derived from the learning process at the component-level, or in this instance, the predictions of the input node:

$$\mu_{i,t} = \hat{\mu}_{i,t}^{\text{input}} \quad (16)$$

$$\tau_{i,t} = \hat{\tau}_{i,t}^{\text{input}} \quad (17)$$

We use  $\tau$  to represent the precision of the representational capacity distribution and  $\pi$  to represent precision within the HGF framework to emphasise the conceptual difference. Where  $\pi$  is understood as an estimate of uncertainty,  $\tau$  is related to the component's system-level representational scope in observation space. Thus, low values of  $\tau$  are not interpreted as the model being uncertain about the representational category. Since  $\tau_{i,t}$  is directly derived from the precision prediction of the input node, whose initial condition is in effect set by  $\sigma$  (as described above), at the system-level we can understand  $\sigma$  as controlling the initial value of  $\tau_{n,t}$ .

The weight  $w_{i,t}$  is a quantity representing the relative magnitude of the  $i$ th component's representational capacity on the system-level. This means that the model can differentially weight components irrespective of their representational scope in observation space (given by  $\theta$ ), thereby allowing for differentiation in the representational space of components that do not differ in terms of  $\theta$ , i.e., that cover the same area of observation space. As we shall see, this is crucial to the adaptability of the structure learning dynamic.

We use the notation  $c_i$  to refer to the  $i$ th component as a whole. At the system-level,  $c_{i,t}$  is the representational capacity given by the combination of  $\theta_{i,t}$  and  $w_{i,t}$ .

### 2.2.2. Activity Level and Weight Update

At each timestep  $t$ , the model is exposed to a new observation  $o_t$ , which constrains the update of the components. Before updating each component, an activity level  $a_{i,t} \in [0, 1]$  is computed. The full set of activity levels at time  $t$  forms a categorical probability distribution over components:

$$\sum_i a_{i,t} = 1 \quad (18)$$

We define each probability  $a_{i,t}$  as the probability of the  $i$ th component given a set of internal and external constraining factors, which can be expressed in the general form:

$$a_i = p(c_i|I, E) \quad (19)$$

Where  $I$  represents some internal constraint posed from within the system, and  $E$  represents an external constraint posed by the environment. At a specific timestep we here keep the internal constraint arbitrarily defined (this is further discussed below) while taking the observation to be the externally constraining factor:

$$E_t = o_t \quad (20)$$

In the following, we use  $o_t$  to denote the external constraint for clarity in relation to our current implementation, where only *sensory* constraints are posed externally. However, theoretically  $E$  could represent any number of external factors, e.g. *physical* or *social* constraints.

To calculate  $a_{i,t}$ , we use Bayes' theorem to decompose the conditional probability into the following proportionality:

$$p(c_{i,t}|I, o_t) \propto p(I|c_{i,t}, o_t) \cdot p(c_{i,t}|o_t) \propto p(I|c_{i,t}, o_t) \cdot p(o_t|c_{i,t}) \cdot p(c_{i,t}) \quad (21)$$

We define the marginal probability of the component in terms of component weight:

$$p(c_{i,t}) = \frac{w_{i,t}}{\sum_j w_{j,t}} \quad (22)$$

We define the probability of the component given the observation with the probability density function for the Gaussian distribution  $f$ :

$$p(o_t|c_{i,t}) = \frac{f(o_t, \theta_{i,t})}{\sum_j f(o_t, \theta_{j,t})} \quad (23)$$

This leaves us to define the term  $p(I|c_{i,t}, o_t)$ . As  $I$  can be arbitrarily defined, we conceptualise this term as a probabilistic expression of a system-wide constraint affecting each component individually, which is in turn locally constrained by the interaction between the given component and the external constraint (here the observation). We formally define it as given by the function  $g$ :

$$p(I|c_{i,t}, o_t) = g(\lambda, o_t, \theta_{i,t}) = \frac{2}{\pi} \arctan\left(\frac{e^\lambda}{(o_t - \mu_{i,t})^2 \tau_{i,t}}\right) \quad (24)$$

where  $\lambda$  is a global model parameter that inversely scales the general level of internal constraint  $I$  across all components, i.e., when  $\lambda$  is large, components are generally less constrained. The term  $(o_t - \mu_{i,t})^2 \tau_{i,t}$  is constructed to express the interaction between the component and the external constraint on which  $I$  is conditioned. This is a modelling choice that ensures that internal constraint varies dynamically with the component-level learning process, such that it remains component-specific rather than being rescaled relative to the full ensemble. The ratio  $e^\lambda / ((o_t - \mu_{i,t})^2 \tau_{i,t})$  is translated

into probability space using  $\frac{2}{\pi} \arctan$ . Here we note that  $\pi$  represents the common constant  $\pi \approx 3.14$  and should be dissociated from the use of the same letter in the HGF equations. Taken together, the function  $g$  is a probabilistic expression of a constraint relation that is more limiting the further away the observation is from the component's concentration of representational capacity in the observation space, where  $\lambda$  scales this constraint. The functional point of this term is that it is a structure independent (i.e. independent of other components) scaling of the relevance of the observation. This is vital for the dynamic as it enables the constraining of components for which the observation is not relevant, and this is the basic mechanism of representational differentiation in the model. In this sense,  $\lambda$  can be understood as controlling the diffusion of potential relevance in the observation space relative to the representational capacity of a component.

Finally we can express the activity level for the  $i$ th component at time  $t$  as:

$$a_{i,t} = \frac{g(\lambda, o_t, \theta_{i,t})f(o_t, \theta_{i,t})w_{i,t}}{\sum_j g(\lambda, o_t, \theta_{j,t})f(o_t, \theta_{j,t})w_{j,t}} \quad (25)$$

At every timestep the weight of each component is updated towards the current activity level scaled by the function  $g$ :

$$w_{i,t+1} = w_{i,t} + g(\lambda, o_t, \theta_{i,t})(a_{i,t} - w_{i,t}) \quad (26)$$

In this way the weight represents a trace of the component's activity history dynamically modulated by the internal constraint and the component's own interaction with the environment. This effectively means that the weight reflects the activity history in proportion to the representational capacity of the component throughout that history.

Components are initialised with the following weight:

$$w_{n,t} = \alpha \quad (27)$$

Here  $\alpha$  represents a global parameter determining both the initial weight of the component and the global threshold for the removal of a component.

### 2.2.3. Adding and Removing Components

Recall that the model's structure evolves continuously as the result of an iterative update cycle, where all components learn in parallel at the component-level. At each timestep, the update cycle includes the following stages in sequence: (1) A new component is added. (2) activity is calculated for all components. (3) HGFs and weights are updated for all but the new component. (4) Components with insufficient weight are removed. At the system-level, since the constraint relations that we model are understood as momentary, the full update cycle is viewed as a model of a single event. Thus, a component that did not survive its initial update cycle can be understood as an unrealised potential. It could have come into existence during the course of the event, but it didn't. As such, the form of the new component, as computationally initialised, reflects the model's momentary propensity for model expansion, where the form it takes at the end of the update cycle represents the potentially realised component. Thus, when we refer to a *new* component we refer to the computational entity playing the role of newly added component within the computational procedure, which are understood as something not realised in the system we model.

The activity level and weight for a newly added component are computed differently. Per definition, when  $\mu_{n,t}^{\text{value}} = o_t$  (as it is at initialisation), then necessarily  $o_t - \mu_{i,t} = 0$ , which leaves the function  $g$  in Equation (24) undefined due to division by zero. While it might seem sensible to define this function to be 1 for new components, as  $g(\lambda, o_t, \theta_{i,t})$  approaches 1 when  $(o_t - \mu_{i,t})^2$  approaches 0, there is a conceptual gap between viewing components at this stage as yet *unrealised* and viewing this term as incorporating a *realised* interaction between the component and the environment. For this reason, and in line with interpretation of  $\alpha$  (see below), we define  $p(I|c_{n,t}, o_t)$  for new components as:

$$p(I|c_{n,t}, o_t) = \alpha \quad (28)$$

Otherwise, new components take part in the calculation of activity levels as if they were an established component. However, for the new component there is no realised learning process at the component-level affected by the activity level, which is why the update step for these are skipped for new components. Instead, the weight for the next time step, i.e. the weight of the component if realised, is directly set to its activity level:

$$w_{n,t+1} = a_{n,t} \quad (29)$$

Components with  $w_{i,t+1} < \alpha$  at the end of the update cycle are removed permanently from the model. This ensures that only new components that gain sufficient weight during their initial update cycle are effectively added to the model, marking its realisation. This prevents the model from continuously accumulating components, while retaining the ability to dynamically expand in direct response to the environment. The larger  $\alpha$  is, the more likely the new component is to be realised, relative to the representational capacity of other components at the same location in observation space. In this sense,  $\alpha$  represents a general pressure to expand the model structure, which is constrained by the representational capacities already captured by the model.

Adding and removing components, allows for both structural expansion and contraction. However, it is through the unfolding of the temporally extended dynamics that this becomes structure learning. Here, the notion of *tension* between components plays an important explanatory role. We briefly outline some of the dynamics related to structural expansion and contraction.

### 2.3. Tension and Contraction

The discrepancy between  $g(\lambda, o_t, \theta_{i,t})$  (Equation (24)) and the activity level (Equation (25)) is an especially important relation, driving the flexible development of the model. Since one is structurally dependent and the other structurally independent, a discrepancy such that the value of  $g$  is large and the activity level is small means that there are other components with representational capacity at the same location. This creates a tension between components. As all involved components will update their weights towards their respective activity level, relatively unconstrained by  $g$  (due to its value being large), if observations recur at roughly the same location over time, components with lower weights will lose weight faster than others and will eventually either perish or be attracted to another location in observation space. This either resolves the tension with a remaining component dominating the given location (regaining weight on subsequent timesteps), or stabilises the tension between the internal and external constraints, allowing for shared representational capacity in the given area of the observation space.

The realisation of this dynamic depends on  $\lambda$ , as this parameter effectively controls  $g$ 's (Equation (24)) sensitivity to representational capacity. For extremely large values of  $\lambda$  the value of  $g$  will always be near 1, and conversely for an extremely low  $\lambda$  values the value of  $g$  will always be near 0. This means that the effect of  $\lambda$  falls on a continuum, where at one end (large values) *all* component weights update maximally every timestep, i.e. becoming practically equal to the current activity level, and as  $\lambda$  decreases weight updating is more and more suppressed. For the structure learning dynamic this means that  $\lambda$  scales the limits of sustained tension between components, and consequently controls how much shared representational capacity or representational ambiguity is tolerated.

### 2.4. Parametrisation and Expansion

Since the representational capacity of new components is centred on the observation, the more precise the distribution is (i.e. larger  $\tau_n$ ), the larger the  $f$  (Equation (23)) is in the calculation of the activity level (Equation (25)). This means that  $\sigma$  effectively scales the activity level of new components, such that smaller  $\sigma$  values amplify the propensity for structural expansion. Consequently, low  $\sigma$  values may result in elevated structural expansion of overly precise components, which due to their high

concentration of representational capacity result in less tension between components, making it less likely that the dynamic will resolve into a proper representational structure as described above.

Low values of  $\lambda$  also interfere with the propensity for structural expansion, but in a sense, for the opposite reason of that described for  $\sigma$ . Recall that low values of  $\lambda$  suppress weight updating. This is because the function  $g$  returns lower values when  $\lambda$  is decreased, but since  $g$  is not involved in the calculations for new components,  $\lambda$  has no suppressing effect here. Consequently, low  $\lambda$  values elevate the propensity for structural expansion by suppressing everything but the new component, resulting in greater activation levels and thus greater initial weight for new components.

### 2.5. Simulation Experiments and Measures

We run a series of simulation experiments to test our framework. In these experiments we investigate various aspects of the model, in terms of environmental alignment from a behavioural perspective. Such a behavioural perspective, i.e. forcing the model to classify what component represents a given observation, provides a simple way to measure structural alignment.

In all simulations, the environment consists of three environmental components of standard deviation 1, with equal distance to neighbouring components. The difference between environments, i.e. different levels of ambiguity, is given by the distance between neighbouring means. When this distance is 6, we call it unambiguous, as there is virtually no overlap. When the distance is 3 there is a considerable overlap and we call it ambiguous. When the distance is 1 we call it very ambiguous (see Figure 2A for a graphical representation).

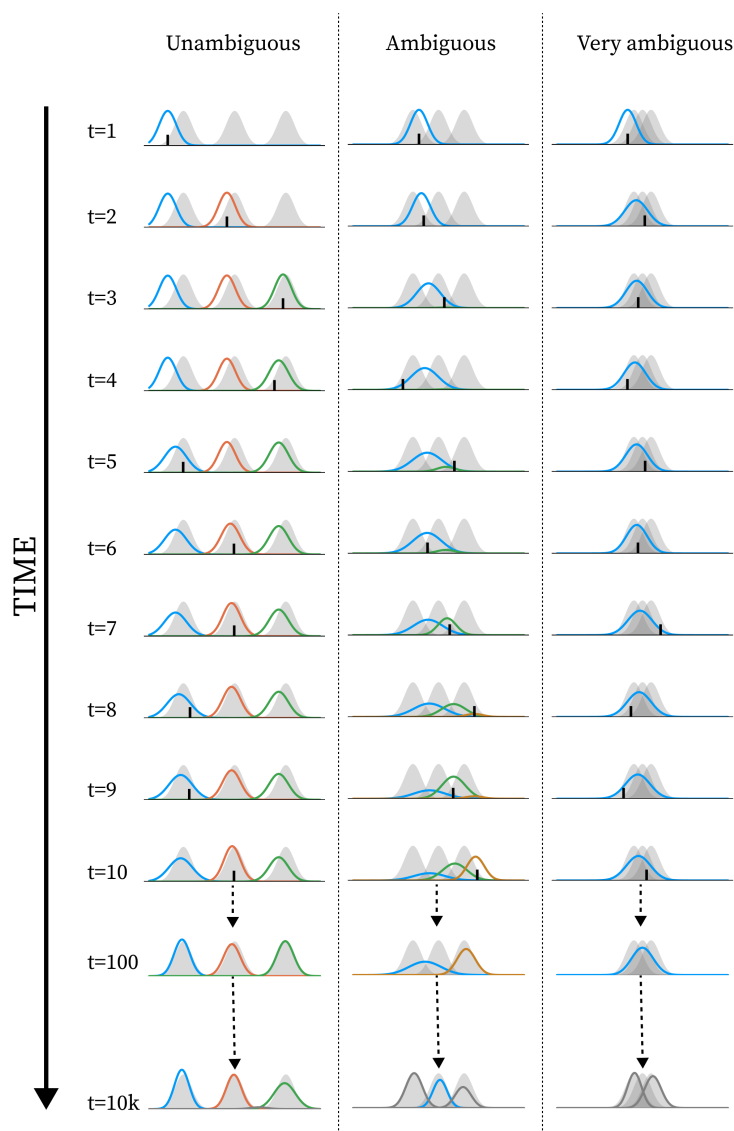
To measure the alignment between the structure of the environment and the structure learned by a model from a behavioural perspective, we use the adjusted Rand index [23] between a model's judgement of a set of observations (i.e. judgment about what "thing" generated them) and the environmental ground truth. The Rand index is a measure developed for testing the similarity of two ways of clustering the same dataset, which is agnostic to the number of clusters in each. For the adjusted Rand index a value of 0 indicates that the similarity is at chance level and a value of 1 indicates that they are identical. To get behavioural data from the model, we draw a sample from a categorical distribution parametrised by the distribution of activity levels calculated for a given observation. This sample serves as the model's judgment about what "thing" generated the observation, and for a set of observations, such judgments correspond to a clustering of the observations. When calculating the adjusted Rand index, we keep the model fixed (i.e. preventing updating) and test the given state of the model for a set of observations. As such, we use a training and test phase approach. In this way the adjusted Rand index serves as a measure of a model's potential behavioural performance for a given state.

In the training phase the model simulation is run for a sequence of timesteps as described in the model description. In the test phase, 20 random observations are drawn from each environmental component. For each of these observations, a distribution of activity levels is calculated and used as parameters for a categorical distribution from which a sample is drawn. This sample functions as the model's judgment about which component generated the observation. Then for each simulation, the adjusted Rand index between the set of model judgements and the ground truth is calculated as a measure of the model's potential behavioural performance.

## 3. Results

In this section we show a series of simulation results produced by running the structure learning algorithm for different models (i.e. different parametrisations) in different environments. First, in Figure 3 we show examples of how the structure learning play out at the system-level in different environments using the standard parametrisation given above. For a more comprehensive impression, we highly encourage the reader to find animations of such examples in the supplementary material or try out the interactive simulation tool built to support the communication of the model dynamics by visiting <https://ilabcode.github.io/constraint-based-structure-learning/>. We then investigate the

effects of  $\sigma$  at various levels, before doing the same for  $\lambda$  and finally we investigate the interaction between the two.

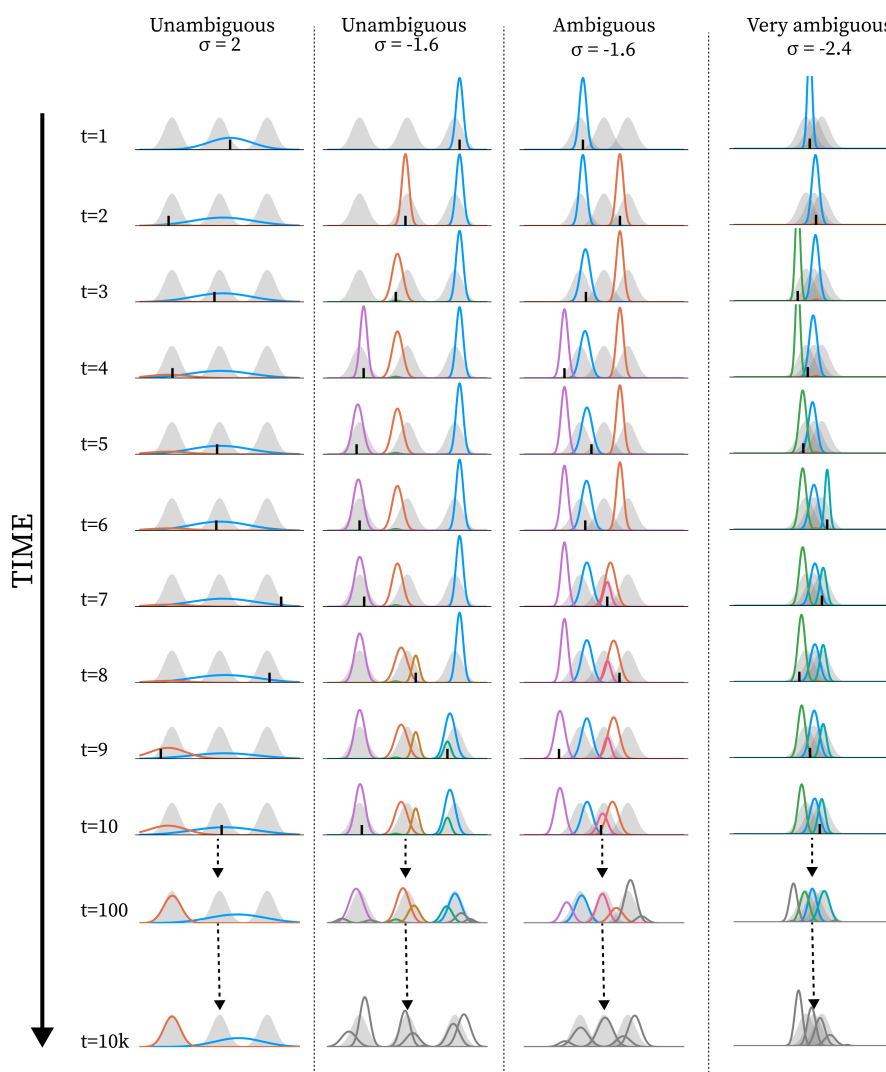


**Figure 3.** Examples of how the learning dynamic unfolds with standard parametrisation at different levels of ambiguity. The 3 columns of plots each represent a simulation spanning 10,000 timesteps. Every row represents a timestep denoted to the right of the row. For the first 10 rows we show the initial learning during the first 10 timesteps. The last two rows represent the state of the model after 100 timesteps and at the end of the simulation. The x-axis of every plot represents the observation space. Every plot shows the environmental components as grey shaded Gaussian distributions. Each coloured line represents the distribution of representational capacity for a model component, where the height is scaled by the weight of the component. Each plot shows the state of the model after updating on the given timestep and for the first 10 timesteps a small black line represents the observation that these updates were based on. The colour of the lines represents the timestep the component was added within the first 10 timesteps. Components added after the first 10 timesteps are shown in grey. The leftmost column shows a simulation in an unambiguous environment (distance between means = 6), the middle column in an ambiguous environment (distance between means = 3), and the rightmost column in a very ambiguous environment (distance between means = 1). The examples are chosen to show typical simulation runs.

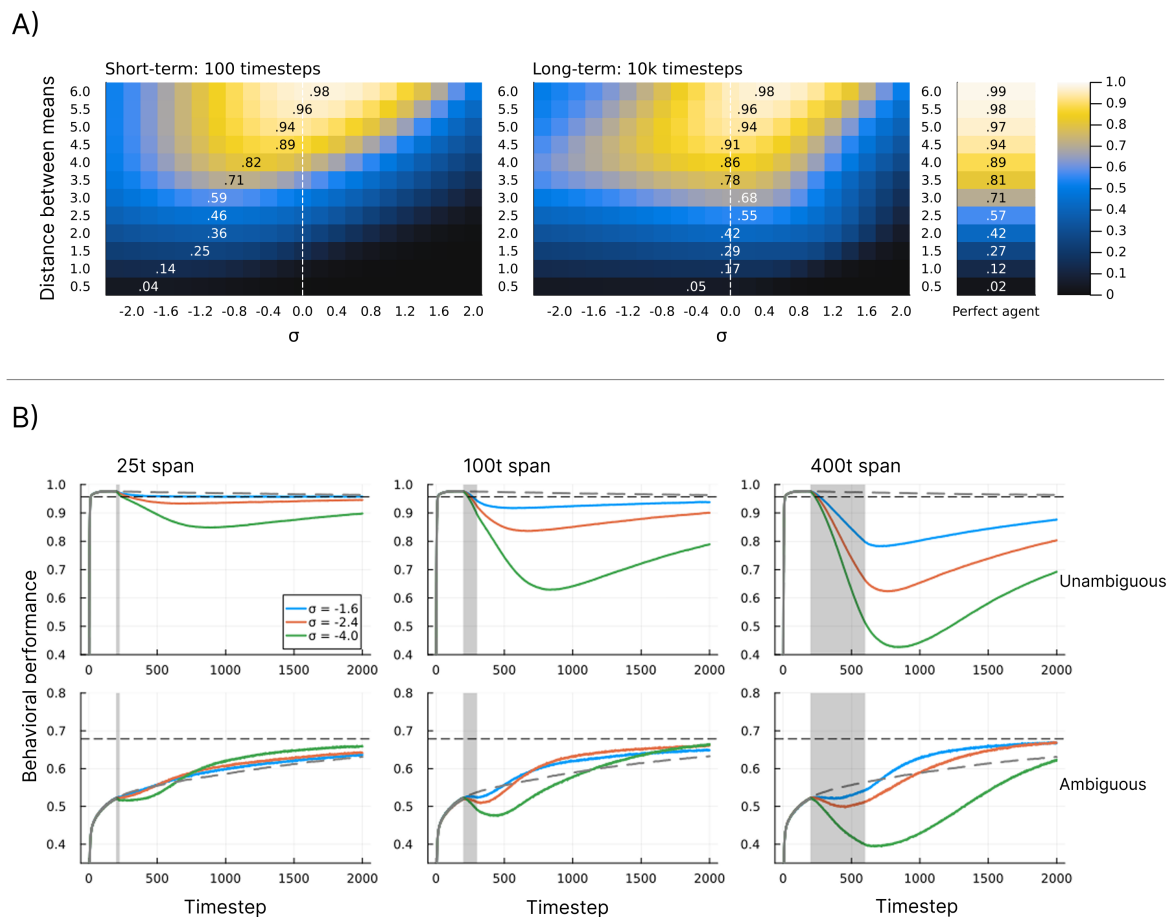
### 3.1. Investigating $\sigma$

In Figure 4 we see examples of how the structure learning dynamic might unfold at various values of  $\sigma$  diverging from the standard parametrisation. In Figure 5A we show the behavioural performance

across a range of  $\sigma$  values and at different levels of environmental ambiguity. Here we see that the optimal value of  $\sigma$  roughly corresponds to the value matching the precision of the environmental components. However, for short-term learning, as the environment becomes more ambiguous, the optimal  $\sigma$  value shifts downwards, indicating that higher initial precision is preferred in ambiguous environments. In the case of long-term learning, these optima remain close to the  $\sigma$  value matching the environment, and here they are notably closer to the performance of perfect agents, i.e. agents with a component structure identical to the environment. In Figure 5B we see that in unambiguous environments the models reach good average performance quickly and then slowly decrease towards a slightly lower convergence point. Here we see that decreasing  $\sigma$  within a time span has a negative effect on performance, and the magnitude of this effect scales with both the level of  $\sigma$  and the length of the span. We also see that the duration of these effects extends beyond the span, with trajectories following a continued decrease followed by a slow shift towards an increasing trend. For ambiguous environments, we see that models have a much slower increase towards the convergence point under standard parametrisation. Here, we see a similar pattern of negative effects for decreasing  $\sigma$ , but only within the time span. After the span, the effect is an overall increase in how fast the models approach the convergence point on average, with some combinations of  $\sigma$  values and span length being better than others.



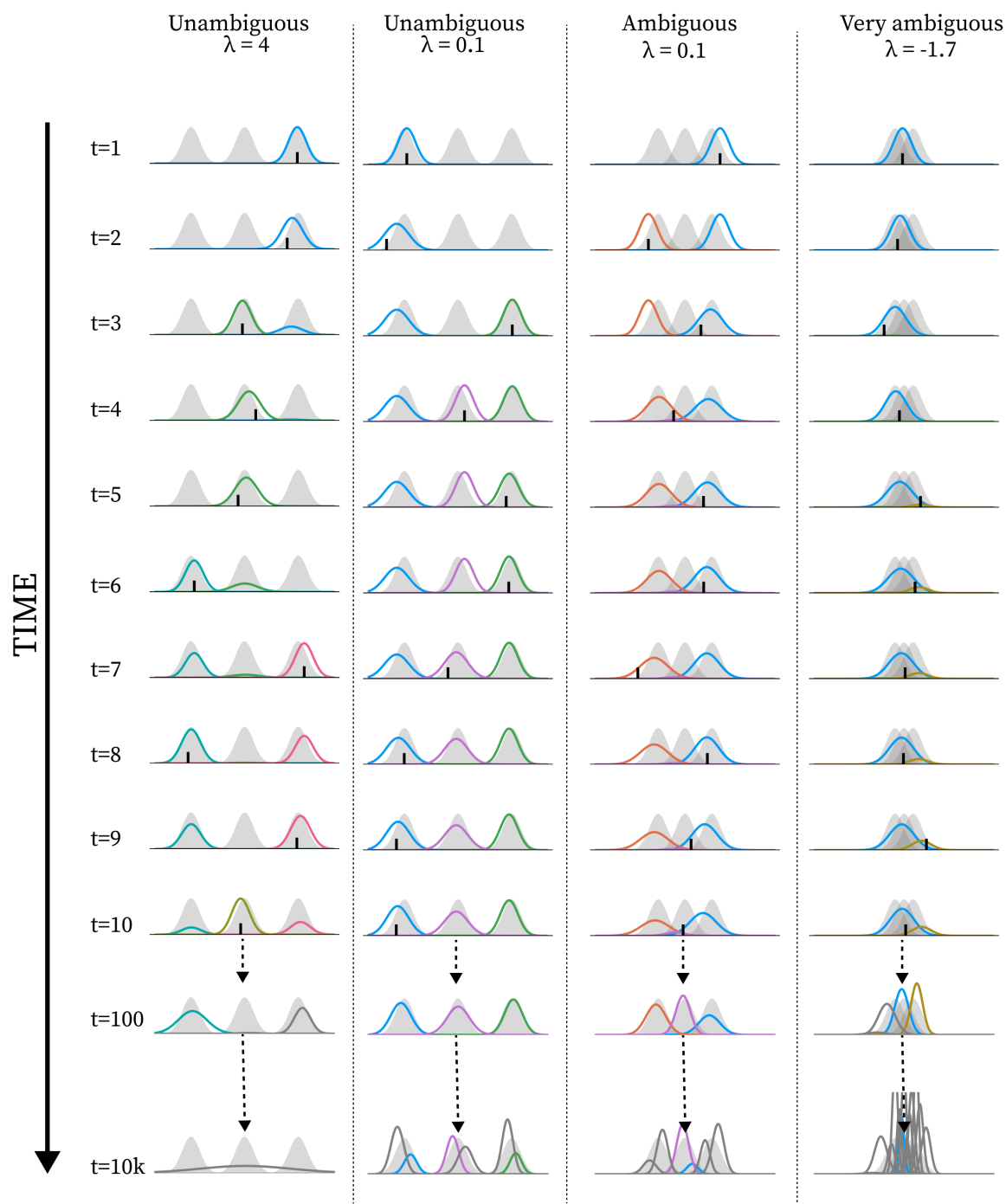
**Figure 4.** Examples of how the learning dynamic unfolds with different levels of  $\sigma$  at different levels of ambiguity. The plot is constructed in the same way as described in Figure 3, except that there are four columns, with the level of ambiguity and the level of  $\sigma$  used denoted at the top of each column.



**Figure 5.** Analysis of behavioural performance for various manipulations of  $\sigma$ . **A)** Heatmaps showing model performance under varying  $\sigma$  values and levels of ambiguity. Each cell depicts the mean behavioural performance across 10,000 simulations using a  $\sigma$  value represented by the x-axis and a distance between neighbouring means given by the value on the y-axis. The mean behavioural performance is represented by a colour given by the colour gradient shown in the rightmost legend. Each simulation runs for a number of timesteps in a training phase, before behavioural performance is calculated in a test phase. One heatmap shows short-term results (a training phase of 100 timesteps) and the other shows long-term results (a training phase of 10,000 timesteps), as denoted above the heatmaps. The numbered cells represent the maximum performance value of the given row. The dashed line signifies the  $\sigma$  level that corresponds to new model components matching the precision of the environmental components. The column to the right depicts the results of the same simulation procedure, but using a model with components perfectly mirroring the environmental components. Note that a perfect agent is not perfect in the sense of perfect judgement, as this is in principle impossible in ambiguous environments. **B)** Six plots showing the average trajectory of behavioural performance. In each plot, for each of four tests, 10,000 simulations are run for 2000 timesteps. At each timestep, behavioural performance is measured with a test phase after the update cycle. Each line represents average performance across all simulations (y-axis) at each timestep (x-axis). The grey lines (dashed after  $t = 200$ ) represent tests using the standard parametrization throughout. Other lines represent tests where parametrization is non-standard within a time span. The grey box represents the time span (starting at  $t = 200$ ) during which the parametrization diverges from the standard. Each column of plots represents a span denoted at the top of the column (spans of 25, 100, and 400 timesteps). For blue lines,  $\sigma = -1.6$  within the span; for red lines,  $\sigma = -2.4$ ; and for green lines,  $\sigma = -4$ . For all simulations,  $\sigma = 0$  before and after the span. The black dashed line represents the convergence point under standard parametrization, i.e. the average performance after 100,000 timesteps. The top row of plots represents tests in unambiguous environments (distance between means = 6), and the bottom row represents tests in ambiguous environments (distance between means = 3).

### 3.2. Investigating $\lambda$

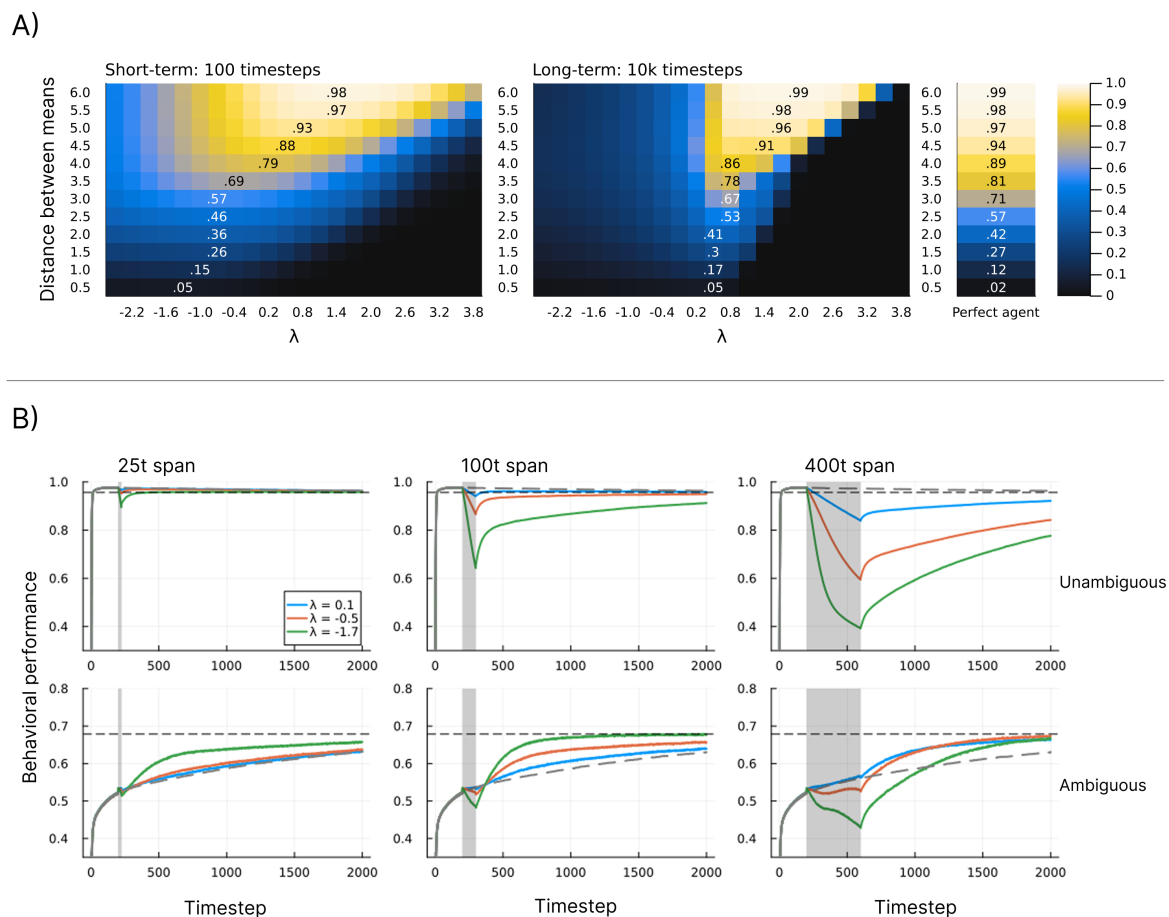
In Figure 6 we see examples of how the structure learning dynamic might unfold at various values of  $\lambda$  diverging from the standard parametrisation. Notice here how the over- and under-differentiation of the observation space is similar to that observed for varying  $\sigma$  values in the long term, but not in the short term (see Figure 4).



**Figure 6.** Examples of how the learning dynamic unfolds with different levels of  $\lambda$  at different levels of ambiguity. The plot is constructed in the same way as described in Figure 3, except that there are four columns, with the level of ambiguity and the level of  $\lambda$  used denoted at the top of each column.

In Figure 7A we see the same general pattern for varying  $\lambda$  values in the short term as for  $\sigma$  values in Figure 5A. For long-term simulations,  $\lambda$  values lower than the standard parametrisation ( $\lambda = 0.7$ )

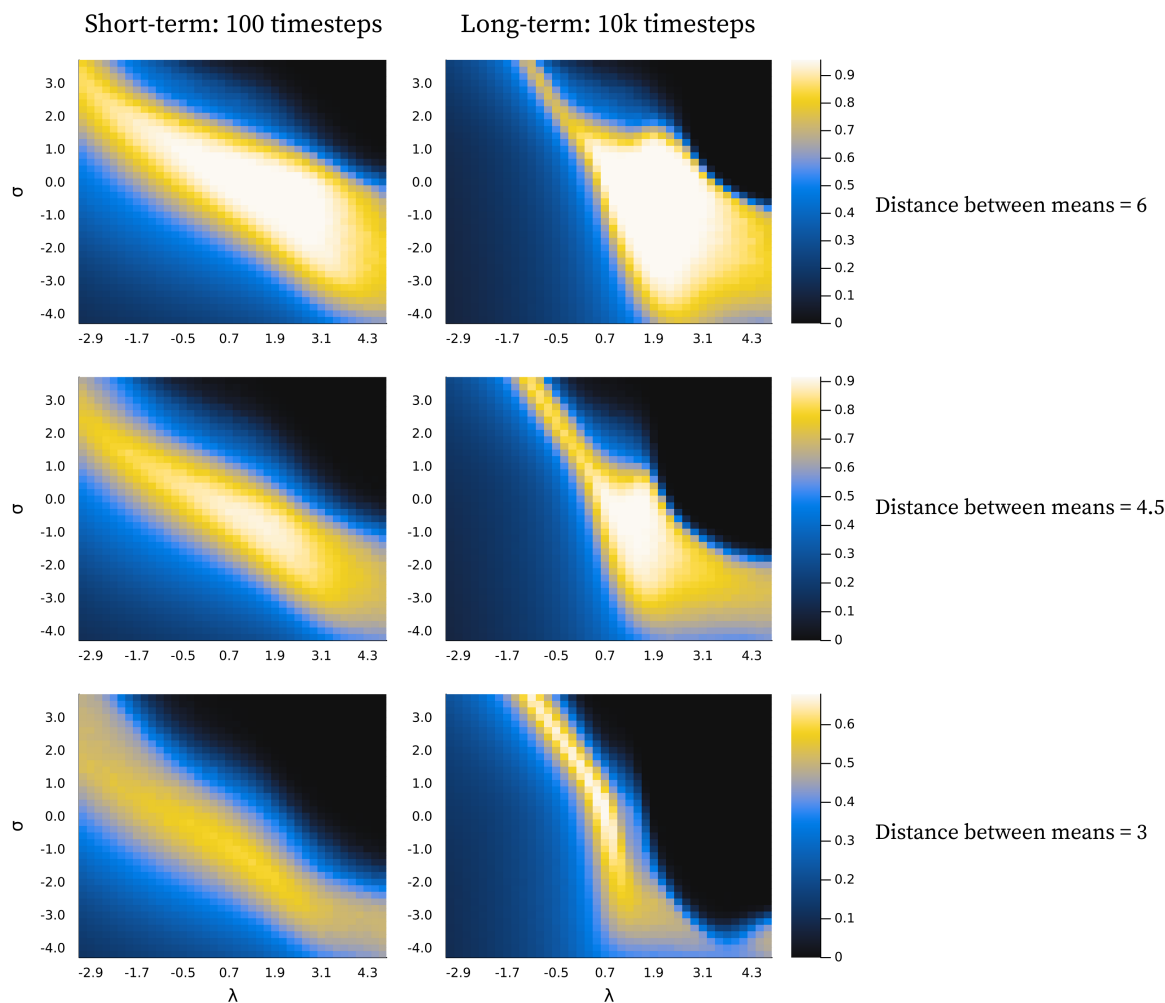
result in worsened performance, and for higher values there is a sharp decrease in performance at specific values depending on the level of ambiguity. In Figure 7B we also see the same general pattern as for  $\sigma$  in Figure 5B, with the noticeable difference that the shape of the curves is inverted. Here, the decrease and increase in average performance are more rapid at the start and end of the span, respectively, and the shift is instantaneous at the end of the span.



**Figure 7.** Analysis of behavioural performance for various manipulations of  $\lambda$ . All plots are produced in the same way as in Figure 5, except that varying  $\lambda$  values are used and  $\sigma = 0$  across all simulations. In **B)**, blue lines correspond to  $\lambda = 0.1$  within the span, red lines to  $\lambda = 0.5$ , and green lines to  $\lambda = -1.7$ . For all simulations,  $\lambda = 0.7$  before and after the span.

### 3.3. Investigating Interaction Between $\sigma$ and $\lambda$

In Figure 8 we show that there is a negative relationship between  $\sigma$  and  $\lambda$  in terms of optimal behaviour. This means that, to some degree, one parameter can compensate for the behavioural effects produced by the other, with some areas of parameter space being generally better. We also clearly see that the standard parametrisation lies within the preferable range across all conditions shown.



**Figure 8.** Model performance under varying  $\sigma$  and  $\lambda$  values. Each cell represents average behavioural performance, produced and depicted in the same way as in Figure 5A and Figure 7A. The x-axes represent varying  $\lambda$  values and the y-axes represent varying  $\sigma$  values. Each row of plots represents a different environment, denoted to the right of the row. Note that each row uses a different mapping to the colour scale, and therefore colours should not be compared across rows. The convergence point under standard parametrisation for the given environment (i.e. the average behavioural performance after 100,000 timesteps) is used as the top value of the colour scale. This is done to more clearly depict the shape of the distribution in each plot.

The difference between the best performance of the trained models and the constructed perfect agent (agents with a component structure identical to the environment) in Figure 5A and Figure 7A is very similar across all levels of ambiguity in the long term. Note that here, *perfect* does not refer to perfect classification. In ambiguous environments, this would be possible only by chance and would not suffice as a measure of perfect learning. Instead, it refers to an agent with a set of components where  $\theta$  exactly corresponds to the environmental components, i.e. an agent that perfectly represents the environment. As such, the results from the perfect agent provide a benchmark, representing the best performance we can expect the model to achieve under the given conditions. Interestingly, in highly ambiguous environments the optimum for the trained models is slightly better than that of the perfect agents. However, performance for both the trained models and the perfect agents is close to chance level, and the slight advantage may be an artefact of the relatively simplistic measure used. In the short term, as the environment becomes more ambiguous, the average performance difference increases between the optimum for the trained models and the perfect agents. This is due to a slower trajectory towards the convergence point, as exemplified by the grey lines in Figure 5B and Figure 7B.

In Figure 5A and Figure 7A we see the same general pattern for varying  $\sigma$  and  $\lambda$  values in the short term. However, this pattern emerges for slightly different reasons. Both show worse performance for lower values in unambiguous environments because too many components are present, i.e. the observation space is over-differentiated. This is partially due to both parameters indirectly affecting how easily new components enter the model. However, in the case of small  $\lambda$  values, the model remains in this state because as  $\lambda$  decreases it becomes harder to eliminate inappropriate new components due to increased constraints on weight updating. In the short term, for ambiguous environments, the lower optimum for  $\lambda$  arises because lower  $\lambda$  values allow the model to tolerate a higher level of ambiguity, whereas in the case of  $\sigma$ , the lower optimum is due to its ability to differentiate with higher precision. However, as exemplified in Figure 4 and Figure 6, consistently low values of both parameters almost always result in gross over-differentiation in the long term.

Our results show that under certain parametrisations (e.g. low  $\sigma$  or low  $\lambda$ ), the model is likely to develop representations that do not align with any component of the environment. Examples of this can most clearly be seen in Figure 4 as well as in the animations provided in the supplementary material. We also observe components that remain relatively fixed over some time periods, especially for low  $\lambda$  values, as this is one of the direct effects of decreasing  $\lambda$ . In the next section we briefly describe how this mechanic can be used to explain maladaptive structure learning.

### 3.4. Modelling Delusions as Maladaptive Belief Updating

Delusions are commonly defined as fixed, false beliefs held with high conviction and resistant to counter-evidence [24]. Computational accounts therefore often frame delusions as maladaptive belief updating, but open questions remain about how misaligned beliefs *emerge* and how they become *maintained* or entrenched over time [25]. Our framework offers an intuitive account of the emergence of delusions, whereby through exposure to ambiguous environments, constraint-based dynamics can stabilise representations that are persistently misaligned with the true environmental components. In our simulations, both low initial component precision ( $\sigma$ ) and low internal constraint ( $\lambda$ ) promote the formation of overly narrow, ad hoc categories, resulting in fixed misalignment. Within this view, delusions arise not from an aberrant processing of prediction errors, but as emergent from system-level constraints that modulate learning and competition among candidate representations.

## 4. Discussion

The simulation results provide a general proof of principle that our model dynamic can perform structure learning in simple environments under the standard parametrisation given above. This capacity to produce properly differentiated internal categories from undifferentiated observations supports the paper's core claim, namely that representational structure can emerge from constraint-based dynamics. Additionally, we explored several aspects of the model dynamic related to interactions between regions of parameter space and environmental configuration. In this section, we discuss these aspects further and outline limitations and future directions for this constraint-based approach.

The model shows robust behavioural performance in unambiguous environments across a range of  $\sigma$  and  $\lambda$  values (Figure 8). In ambiguous environments, the behavioural optimum lies at lower values of  $\sigma$  (Figure 5A) and  $\lambda$  (Figure 7A) for short-term learning, but less so for long-term learning. However, as seen in Figure 5B and Figure 7B, a short-duration decrease in either parameter value may aid long-term learning by accelerating structure learning overall.

A decrease in both  $\sigma$  and  $\lambda$  is associated with a risk of representational over-differentiation (for different reasons). In Figure 8, the black area in the top right corner represents a region of parameter space that often results in under-differentiation. Since these parameter values show the worst performance, under-differentiation may lead to more severe behavioural consequences than over-differentiation. This suggests that, in the face of ambiguity, risking over-differentiation can be advantageous, especially when there is a risk of under-differentiation. Extrapolating from these results, one might expect an adaptive structure learning agent to adjust to novel, transforming, or otherwise informationally obscured environments by increasing the level of structural expansion,

despite the risk of over-differentiation. However, the consequences of behaviour may of course be highly context-dependent. At least intuitively, one might argue that it is easier to apply the same behaviour to perceptually distinct phenomena than to precisely apply different behaviour to the same perceptual phenomenon. In fact, in order to do the latter, some perceptual differentiation would have to exist at some level. According to this logic, in an ambiguous environment it may be safer to over-differentiate in order to retain the option of applying distinct behavioural responses.

It is important to note that the behavioural measure used in this study captures potential behavioural performance at a given time point, but does not reflect structural coherence across time. Considering Figure 8, it may seem preferable to combine a relatively large  $\lambda$  value with a relatively low  $\sigma$  value in unambiguous environments. However, large  $\lambda$  values often come at the cost of components being replaced more frequently (e.g. see the first column in Figure 6). Conversely, high  $\sigma$  values are also associated with similar long-term structural incoherence. In fact, it is extremely rare for components added during the short-term phase (first 100 timesteps) to still be present in the model at  $t = 10,000$ . We find that our standard parametrisation offers a good balance between flexibility and coherence, a compromise that future work should investigate more rigorously.

Our results suggest that both decreases in  $\lambda$  and  $\sigma$  may explain the emergence of maladaptive delusion-like representations, albeit with different representational expressions. One might interpret this as relating to differences in delusional content, but we do not pursue this line of inquiry here. Rather, the emergence of delusions in the model appears to be related to an elevated propensity for structural expansion, that is, how easily new components enter the model. Since both  $\lambda$  and  $\sigma$  are related to this propensity, each may contribute to such outcomes. So far, we have not investigated the role of  $\alpha$  in this context, but it should be noted that  $\alpha$  also affects this aspect, as it controls the baseline propensity for structural expansion.

This provides a novel insights than can explain why delusions emerge gradually and without a clear moment at which the development of a delusional belief coincides with a behavioural shift [26,27]. They may begin as vague background intuitions that only later become fully articulated and behaviourally expressed, making the onset of delusion difficult to detect from a behavioural perspective. This suggests that delusions may be rooted in a developmental process of gradual belief formation, which makes them suitable for modelling within our framework. Indeed, in Figure 5B we see how a perturbation to the structure learning dynamic can alter the learning trajectory with delayed behavioural effects.

Furthermore, following the suggestion that over-differentiation is an adaptive response in ambiguous environments, our results raise the question of whether delusions may be a consequence of such a response in the face of elevated ambiguity. If so, the mechanisms driving the emergence of delusions should not be understood as maladaptive. Instead, the key issue becomes identifying the source of the pressure to adapt when adaptation is not in fact required (e.g. under perceived rather than actual ambiguity). This perspective remains speculative but illustrates a line of inquiry opened up by the present framework.

This work presents a proof of concept rather than a fully fleshed-out framework for modelling cognition. It has not been our intention to provide a competing model, but rather to explore an alternative approach. For this reason, we have not included a comparison with other modelling frameworks, as such a comparison would be beyond the scope of the paper. While the model demonstrates how representational structure can emerge from constraint-based dynamics, several core limitations should be acknowledged.

We explore only two of the four free parameters in the current implementation (three system-level parameters and one component-level parameter). A systematic investigation of interactions across the full parameter space would be too broad for the scope of this paper. Our decision to focus on  $\sigma$  and  $\lambda$  was guided by preliminary analyses suggesting that these parameters reveal particularly informative aspects of the framework. Nevertheless, the full parameter space should be explored in future work.

While we argue for using Bayes' theorem as a probabilistic expression of constraint, not every constraint playing a role in the dynamic is formally modelled in this way. As is evident from the descriptions of various aspects of the dynamics, some constraints emerge as the system evolves. Furthermore, not every formal aspect of the model is explicitly framed in constraint-based terms. At the component level, we use the HGF framework to model Bayesian inference as a learning process. Although the concept of filtering may intuitively relate to constraints, it remains to be clarified how the component level should be interpreted within a fully constraint-based framework if structure learning is to be understood entirely in these terms. Nonetheless, we aim to show that the model is grounded in Bayesian logic expressing a set of core system-level constraints from which structure learning emerges.

We provide no formal proof or other guarantee that the model will converge to a true or optimal structure. Although the model achieves average behavioural performance close to that of a perfect agent under various conditions, this performance measure does not capture how closely model components align with environmental components (e.g. via KL divergence between corresponding distributions). One might expect such alignment to increase in a successful structure learning model, potentially leading to convergence between model and environment over infinite time. Such convergence is not guaranteed here, in part due to the model's dependence on observational sequences. Indeed, the convergence point for unambiguous environments in Figure 5B and Figure 7B is slightly lower than the average performance reached in the short term. This reflects the non-zero probability that an unfavourable sequence of observations will lead to an improper structure, a likelihood that increases over time. We regard this limitation as a strength, as it may better reflect biological and developmental realism. Rather than convergence per se, what matters is whether the structure stabilises in a context-sensitive and dynamically coherent manner.

The model and simulation setup are intentionally simplistic. The observation space is low-dimensional, the environment is static, and there are no behavioural interactions during learning. Extending the model to incorporate behaviour in a dynamic environment as part of the constraint-based dynamics will be a crucial next step. Given that the model builds on constrained spontaneous activity, here implemented as component-level learning processes, it would be natural to extend this framework to behaviour itself. This could involve treating the cognitive system as a mechanism for filtering appropriate behaviour for an otherwise spontaneously behaving agent, potentially integrating behaviour directly into the representational model via component activity levels. Such an approach contrasts with the common separation of learning and response models, in which perception and action are treated as distinct processes. Instead, perception would be modelled as a capacity emerging from constrained spontaneous action.

Future work should also extend the model to incorporate more abstract representations, such as object classes or generalisations. This could perhaps be achieved by performing structure learning in representational space in addition to what is implemented here. One may argue that due to the lack of abstraction as well as the lack of explicit memory, the model presented here is best understood as a model of very low-level processes. In this way, our model may be a good description of an initial dimensionality reduction of complex environmental information, which serves as a representational basis for other mechanisms of abstractions and explicit memory. Future work should explore whether abstractions and memory require specialised mechanisms, or whether they can arise as emergent capacities within constraint-based dynamics, similar to how we have shown that structure learning can emerge.

A central motivation for this work is to explore whether cognitive systems can be computationally modelled as constraint-based systems. In this respect, our simulation results are promising and provide an initial justification for pursuing this approach. One potential advantage of a constraint-based framework is that it may facilitate translation between cognitive and biological processes by allowing both to be described using a shared conceptual vocabulary. Such a framework could help bridge neurobiology and cognitive science, an essential step towards understanding the relationship between

emergent cognitive capacities and measurable neural activity, including the effects of neurobiological interventions on cognition, such as pharmaceutical treatments in psychiatry.

**Supplementary Materials:** For all animation files (.gif), six models are shown. Within the same file, all models receive the same sequence of observations, but they differ in their parametrisation. It is noted above each model animation how it differs from the standard parametrisation. Here, the colour of the components reflects the time during the simulation at which they were added. The colour gradient at the bottom shows the progression of colours across timesteps. The timestep of each frame is also shown. Note that for long-term animations and trajectory animations, only every 10th and 5th update is shown, respectively, to speed up the total animation time. Trajectory animations are intended as examples of the trajectory plots in the article. Here, the background changes to grey during periods of alternate parametrisation. The parameter shown at the top of each simulation denotes the parametrisation used during that time span. When the background is white in the trajectory animations, all models use the standard parametrisation. The file names indicate the content of the animations. Here, DBM denotes “distance between means”. DBM6 corresponds to an unambiguous environment, and DBM3 to an ambiguous environment.

**Author Contributions:** Conceptualization, C.L.O., M.H., C.M. and P.T.W.; methodology, C.L.O., N.L. and C.M.; software, C.L.O.; validation, C.L.O.; formal analysis, C.L.O.; investigation, C.L.O.; resources, C.M.; data curation, C.L.O.; writing—original draft preparation, C.L.O.; writing—review and editing, C.L.O., N.M, M.H., N.L., P.T.W. and C.M.; visualization, C.L.O.; supervision, C.M.; project administration, C.L.O. and C.M.; funding acquisition, C.M. and C.L.O. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by Independent Research Fund Denmark grant number 3166-00158B. This research was funded by Carlsberg Foundation grant number CF21-0439. This research was funded by Wellcome Trust grant number 226776/Z/22/Z. This research was funded by Aarhus Universitets Forskningsfond grant number AUFF-E-2019-7-10.

**Data Availability Statement:** All scripts and data used for the production and analysis of the results are available at [https://osf.io/rmvgb/overview?view\\_only=5b41c73d0eb04337b6947b80256d59e5](https://osf.io/rmvgb/overview?view_only=5b41c73d0eb04337b6947b80256d59e5)

**Acknowledgments:** During the preparation of the simulation tool linked to in this article, the authors used Anthropic’s Claude Opus 4.5 for the purposes of translating the model code used in the article into a working application. The authors have reviewed the output, but note that this tool is developed solely for communication purposes and should not be considered part of the scientific output of this article. No AI have been involved in producing the model code or analysis presented in this article.

**Conflicts of Interest:** The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

## References

1. Tenenbaum, J.B.; Kemp, C.; Griffiths, T.L.; Goodman, N.D. How to Grow a Mind: Statistics, Structure, and Abstraction. *Science* **2011**, *331*, 1279–1285. <https://doi.org/10.1126/science.1192788>.
2. Kemp, C.; Tenenbaum, J.B. The discovery of structural form. *Proceedings of the National Academy of Sciences* **2008**, *105*, 10687–10692. Publisher: Proceedings of the National Academy of Sciences, <https://doi.org/10.1073/pnas.0802631105>.
3. Kitson, N.K.; Constantinou, A.C.; Guo, Z.; Liu, Y.; Chobtham, K. A survey of Bayesian Network structure learning. *Artificial Intelligence Review* **2023**, *56*, 8721–8814. <https://doi.org/10.1007/s10462-022-10351-w>.
4. Gong, T.; Gerstenberg, T.; Mayrhofer, R.; Bramley, N.R. Active causal structure learning in continuous time. *Cognitive Psychology* **2023**, *140*, 101542. <https://doi.org/10.1016/j.cogpsych.2022.101542>.
5. Gershman, S.J.; Niv, Y. Exploring a latent cause theory of classical conditioning. *Learning & Behavior* **2012**, *40*, 255–268. <https://doi.org/10.3758/s13420-012-0080-8>.
6. Neacsu, V.; Mirza, M.B.; Adams, R.A.; Friston, K.J. Structure learning enhances concept formation in synthetic Active Inference agents. *PLOS ONE* **2022**, *17*, e0277199. Publisher: Public Library of Science, <https://doi.org/10.1371/journal.pone.0277199>.

7. Smith, R.; Schwartenbeck, P.; Parr, T.; Friston, K.J. An Active Inference Approach to Modeling Structure Learning: Concept Learning as an Example Case. *Frontiers in Computational Neuroscience* **2020**, *14*. Publisher: Frontiers, <https://doi.org/10.3389/fncom.2020.00041>.
8. Gershman, S.J.; Cikara, M. Social-Structure Learning. *Current Directions in Psychological Science* **2020**, *29*, 460–466. Publisher: SAGE Publications Inc, <https://doi.org/10.1177/0963721420924481>.
9. Tomov, M.S.; Yagati, S.; Kumar, A.; Yang, W.; Gershman, S.J. Discovery of hierarchical representations for efficient planning. *PLOS Computational Biology* **2020**, *16*, e1007594. Publisher: Public Library of Science, <https://doi.org/10.1371/journal.pcbi.1007594>.
10. Van Orden, G.; Hollis, G.; Wallot, S. The Blue-Collar Brain. *Frontiers in Physiology* **2012**, *3*, 207. <https://doi.org/10.3389/fphys.2012.00207>.
11. Ross, L.N.; Jirsa, V.; McIntosh, A.R. The Possibility Space Concept in Neuroscience: Possibilities, Constraints, and Explanations. *European Journal of Neuroscience* **2025**, *61*, e70038. [\\_eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/ejn.70038](https://onlinelibrary.wiley.com/doi/pdf/10.1111/ejn.70038), <https://doi.org/10.1111/ejn.70038>.
12. Winning, J.; Bechtel, W. Rethinking Causality in Biological and Neural Mechanisms: Constraints and Control. *Minds and Machines* **2018**, *28*, 287–310. <https://doi.org/10.1007/s11023-018-9458-5>.
13. Hooker, C. On the Import of Constraints in Complex Dynamical Systems. *Foundations of Science* **2013**, *18*, 757–780. <https://doi.org/10.1007/s10699-012-9304-9>.
14. Montévil, M.; Mossio, M. Biological organisation as closure of constraints. *Journal of Theoretical Biology* **2015**, *372*, 179–191. <https://doi.org/10.1016/j.jtbi.2015.02.029>.
15. Kauffman, S.; Logan, R.K.; Este, R.; Goebel, R.; Hobill, D.; Shmulevich, I. Propagating organization: an enquiry. *Biology & Philosophy* **2008**, *23*, 27–45. <https://doi.org/10.1007/s10539-007-9066-x>.
16. Pattee, H.H. The Physical Basis and Origin of Hierarchical Control. In *LAWS, LANGUAGE and LIFE: Howard Pattee's classic papers on the physics of symbols with contemporary commentary*; Pattee, H.H.; Rączaszek-Leonardi, J., Eds.; Springer Netherlands: Dordrecht, 2012; pp. 91–110. [https://doi.org/10.1007/978-94-007-5161-3\\_6](https://doi.org/10.1007/978-94-007-5161-3_6).
17. Kauffman, S.A.; Kauffman, S.A. *A World Beyond Physics: The Emergence and Evolution of Life*; Oxford University Press: Oxford, New York, 2019.
18. Juarrero, A. *Context Changes Everything: How Constraints Create Coherence*; The MIT Press, The MIT Press: Cambridge, 2023.
19. Deacon, T.W. *Incomplete nature: how mind emerged from matter*, 1st ed ed.; W.W. Norton & Co: New York, 2012.
20. Potter, H.D.; Mitchell, K.J. Beyond Mechanism-Extending Our Concepts of Causation in Neuroscience. *The European Journal of Neuroscience* **2025**, *61*, e70064. <https://doi.org/10.1111/ejn.70064>.
21. Silberstein, M. Context is King: Contextual Emergence in Network Neuroscience, Cognitive Science, and Psychology. In *From Electrons to Elephants and Elections: Exploring the Role of Content and Context*; Wuppuluri, S.; Stewart, I., Eds.; Springer International Publishing: Cham, 2022; pp. 597–640. [https://doi.org/10.1007/978-3-030-92192-7\\_33](https://doi.org/10.1007/978-3-030-92192-7_33).
22. Weber, L.A.; Waade, P.T.; Legrand, N.; Møller, A.H.; Stephan, K.E.; Mathys, C. The generalized Hierarchical Gaussian Filter, 2023. arXiv:2305.10937 [cs] version: 1, <https://doi.org/10.48550/arXiv.2305.10937>.
23. Hubert, L.; Arabie, P. Comparing partitions. *Journal of Classification* **1985**, *2*, 193–218. <https://doi.org/10.1007/BF01908075>.
24. *Diagnostic and statistical manual of mental disorders: DSM-5*, 5th ed ed.; American psychiatric association: Washington, 2013.
25. Erdmann, T.; Mathys, C. A generative framework for the study of delusions. *Schizophrenia Research* **2022**, *245*, 42–49. <https://doi.org/10.1016/j.schres.2020.11.048>.
26. Mourgues-Codern, C.; Benrimoh, D.; Gandhi, J.; Farina, E.A.; Vin, R.; Zamorano, T.; Parekh, D.; Malla, A.; Joobar, R.; Lepage, M.; et al. Emergence and Dynamics of Delusions and Hallucinations Across Stages in Early Psychosis. *Biological Psychiatry* **2025**, *98*, 679–688. <https://doi.org/10.1016/j.biopsych.2025.02.891>.
27. Powers, A.; Angelos, P.A.; Bond, A.; Farina, E.; Fredericks, C.; Gandhi, J.; Greenwald, M.; Hernandez-Busot, G.; Hosein, G.; Kelley, M.; et al. A Computational Account of the Development and Evolution of Psychotic Symptoms. *Biological Psychiatry* **2025**, *97*, 117–127. <https://doi.org/10.1016/j.biopsych.2024.08.026>.

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.