

## Chance-Constrained Active Inference

**Citation for published version (APA):**

van de Laar, T., Senöz, İ., Özçelikkale, A., & Wymeersch, H. (2021). Chance-Constrained Active Inference. *Neural Computation*, 33(10), 2710-2735. [https://doi.org/10.1162/neco\\_a\\_01427](https://doi.org/10.1162/neco_a_01427)

**Document license:**

CC BY

**DOI:**

[10.1162/neco\\_a\\_01427](https://doi.org/10.1162/neco_a_01427)

**Document status and date:**

Published: 01/10/2021

**Document Version:**

Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

**Please check the document version of this publication:**

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

**General rights**

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

[www.tue.nl/taverne](http://www.tue.nl/taverne)

**Take down policy**

If you believe that this document breaches copyright please contact us at:

[openaccess@tue.nl](mailto:openaccess@tue.nl)

providing details and we will investigate your claim.

## Chance-Constrained Active Inference

**Thijs van de Laar**

*t.w.v.d.laar@tue.nl*

*Eindhoven University of Technology, 5612 AP, Eindhoven, The Netherlands*

**İsmail Şenöz**

*i.senoz@tue.nl*

*Eindhoven University of Technology, 5612 AP, Eindhoven, The Netherlands*

**Ayça Özçelikkale**

*ayca.ozcelikkale@angstrom.uu.se*

*Uppsala University, 75237, Uppsala, Sweden*

**Henk Wymeersch**

*henkw@chalmers.se*

*Chalmers University of Technology, 41296, Gothenburg, Sweden*

Active inference (ActInf) is an emerging theory that explains perception and action in biological agents in terms of minimizing a free energy bound on Bayesian surprise. Goal-directed behavior is elicited by introducing prior beliefs on the underlying generative model. In contrast to prior beliefs, which constrain all realizations of a random variable, we propose an alternative approach through chance constraints, which allow for a (typically small) probability of constraint violation, and demonstrate how such constraints can be used as intrinsic drivers for goal-directed behavior in ActInf. We illustrate how chance-constrained ActInf weights all imposed (prior) constraints on the generative model, allowing, for example, for a trade-off between robust control and empirical chance constraint violation. Second, we interpret the proposed solution within a message passing framework. Interestingly, the message passing interpretation is not only relevant to the context of ActInf, but also provides a general-purpose approach that can account for chance constraints on graphical models. The chance constraint message updates can then be readily combined with other prederived message update rules without the need for custom derivations. The proposed chance-constrained message passing framework thus accelerates the search for workable models in general and can be used to complement message-passing formulations on generative neural models.

## 1 Introduction

---

Similar to biological agents, learning to make decisions based on observations and feedback from the environment is an essential task for autonomous artificial agents. Traditionally, adaptive linear control and model predictive control have been successfully applied in this area (Borrelli, Bemporad, & Morari, 2017). Over the past few years, reinforcement learning has become the predominant approach (Recht, 2019). An emerging alternative perspective to decision making under uncertainty is *active inference* (ActInf) (Friston, 2010), a neuroscience-based theory that has been used extensively to explain behavior of biological agents in dynamic environments (Friston, 2010).

ActInf is based in the *free energy principle* (FEP) and postulates that perception and action in biological agents minimize a free energy bound on Bayesian surprise. The free energy is an information-theoretic measure that bounds the current and the future expected statistical surprise, that is, how unpredictable the observations are under a given generative model (GM). The free energy is associated with the Kullback-Leibler (KL) divergence (i.e., the distance) between the approximate and the true posterior. In particular, according to the free energy principle, the agent acts in such a way as to minimize a free-energy bound on the surprise—a Bayesian surprise that, informally speaking, provides a quantification of the difference between the agent's predictions about the system behavior and the observed system behavior. Minimization of free energy is closely related to variational Bayesian methods, reinforcement learning (Sallans & Hinton, 2001; Tschantz, Millidge, Seth, & Buckley, 2020; Sajid, Ball, Parr, & Friston, 2021), and deep generative models (Ueltzhöffer, 2018; Fountas, Sajid, Mediano, & Friston, 2020), another set of popular machine learning approaches (Goodfellow et al., 2014). ActInf is closely related to message passing on graphical models (de Vries & Friston, 2017; Friston, Parr, & de Vries, 2017), and several widely used message passing algorithms, including (loopy) belief propagation, variational message passing, and expectation propagation, can be derived as fixed-point equations of the (Bethe) free energy (Heskes, 2003; Yedidia, Freeman, & Weiss, 2005; Dauwels, 2007; Zhang, Wang, Fettweis, & Gao, 2017). This relation has been harnessed to develop elegant automated methods for ActInf (Schwöbel, Kiebel, & Markovic, 2018; van de Laar & de Vries, 2019).

In addition to investigation of motivating connections with the behavior of the biological systems (Friston, Kilner, & Harrison, 2006; Ramstead, Badcock, & Friston, 2018), ActInf has been successfully utilized in applications in the traditional stochastic control scenarios, such as linear quadratic gaussian (LQG) control and similar standard problems such as maze problems (Hoffmann & Rostalski, 2017; Ueltzhöffer, 2018; Schwöbel et al., 2018; Baltieri & Buckley, 2019; Millidge, Tschantz, Seth, & Buckley, 2020;

Imohiosen, Watson, & Peters, 2020), and exploration-exploitation balancing in multiarmed bandit problems (Markovic, Stojic, Schwoebel, & Kiebel, 2021).

Despite these promising developments, the ActInf framework lacks certain desirable features present in model predictive control. In particular, there is no off-the-shelf standard ActInf formulation that allows inclusion of chance constraints in the problem setting. Chance constraints provide an attractive approach for online decision making for uncertain systems (Mesbah, 2016), where the dynamics are not fully known or the system contains certain components that are best modeled in a stochastic manner. In such settings, constraints on the system behavior, such as the agent remaining in a given region of the environment with a given probability, cannot directly be encoded in terms of prior beliefs. In contrast to approaches that constrain all realizations of the random variables, chance constraints allow for a (typically small) probability of constraint violation, which can significantly improve performance since chance constraints enable the decision maker trade performance with probability of constraint violation (Blackmore, Ono, & Williams, 2011).

This letter proposes a computationally tractable approach to chance-constrained decision making and applies it to an ActInf context. We include chance constraints in the ActInf objective (i.e., the free energy) by using the Lagrangian formalism. We then solve the Lagrangian optimization problem by variational calculus. Finally, we show that the proposed solution not only leads to a modular and scalable message passing framework for ActInf problems but also provides a general-purpose message passing framework that can account for chance constraints on graphical models in general. We claim the following main contributions:

1. We show that the analytic solution to the chance-constrained free energy problem yields posterior beliefs in the form of truncated mixtures. (Theorem 1)
2. We show how this solution can be interpreted in terms of message passing on a factor-graph representation of the generative model. (Theorem 2)
3. Consequently, our results provide a message passing framework that is specifically designed to account for chance constraints.

Message passing is inherently modular, and (variational) message update rules can be prederived and stored in a lookup table for later use (Korl, 2005; van de Laar, 2019). The chance-constrained message updates can then be readily combined with these pre-derived rules without the need for laborious derivations. Our results illustrate that the proposed framework can successfully find solutions so that the rate of constraint violation specified in the original problem and the one that is actually observed during the closed-loop operation is close. The results also illustrate how to balance

the constraints on the actions and the states through the use of a tuning parameter, which enables exploration of different trade-offs between immediate and delayed intervention.

## 2 Problem Statement

---

We start by defining a general factorized generative model  $f$  with respect to an (arbitrary) collection of variables  $x$ . As a notational convention, individual variables are indexed by  $i, j \in \mathcal{V}$ , and factors by  $a, b, c \in \mathcal{F}$  unless stated otherwise. The model then factorizes as

$$f(x) = \prod_{a \in \mathcal{F}} f_a(x_a), \quad (2.1)$$

with nonnegative real functions  $f_a$  and where  $x_a \subset x$  collects the arguments of  $f_a$ . In a probabilistic generative model, the individual factors usually represent conditional probability distributions. Probabilistic inference is then concerned with obtaining an (approximate) posterior belief  $q_j(x_j) \propto \int f(x) dx_{\setminus j}$  over a variable of interest  $x_j$ , where  $x_{\setminus j}$  indicates the integration over all model variables except  $x_j$ .

We now briefly recap how the computation of these beliefs can be performed efficiently and automated over a factor graph (Loeliger, Dauwels, Koch, & Korl, 2004), and how this process can be interpreted as a Bethe free energy minimization problem (Yedidia et al., 2005). With these concepts firmly in place, we move to chance constraints and the formal problem statement in section 2.4.

**2.1 Factor Graphs for Marginal Belief Computation.** A factor graph can be used to visually represent a factorized function. In this letter, we use the bipartite factor graph representation. A bipartite factor graph,

$$\mathcal{G} = (\mathcal{F}, \mathcal{V}, \mathcal{E}),$$

consists of variable nodes  $\mathcal{V}$ , factor nodes  $\mathcal{F}$ , and edges  $\mathcal{E}$  that connect variable nodes with factor nodes. A variable node  $i \in \mathcal{V}$  is connected to a factor node  $a \in \mathcal{F}$  by an edge  $(i, a) \in \mathcal{E}$  if (and only if) the variable  $x_i$  is an argument of the factor-function  $f_a$ . An example section of a graph is drawn in Figure 1, where the circle and square represent a variable and factor node, respectively. We write the neighborhood of a variable node  $i$  as  $\mathcal{F}(i)$ , which collects all factor nodes in  $\mathcal{F}$  that are direct neighbors of  $i$ . Similarly,  $\mathcal{V}(a)$  collects all variable nodes in  $\mathcal{V}$  that are direct neighbors of  $a$ .

Suppose we are interested in obtaining a posterior belief  $q_j(x_j)$ . The belief propagation algorithm (Pearl, 1982) then prescribes that we send messages

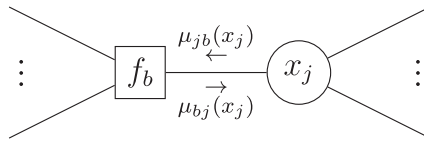


Figure 1: Bipartite subgraph of a model around a variable node  $j$  (circle) and factor node  $b$  (square), with indicated messages. Ellipses represent a continuation of the model.

from the branches of the graph toward the variable node of interest, following the recursive application of the belief propagation update rules,

$$\mu_{jb}(x_j) = \prod_{\substack{a \in \mathcal{F}(j) \\ a \neq b}} \mu_{aj}(x_j), \tag{2.2a}$$

$$\mu_{bj}(x_j) = \int f_b(x_b) \prod_{\substack{i \in \mathcal{V}(b) \\ i \neq j}} \mu_{ib}(x_i) dx_{b \setminus j}, \tag{2.2b}$$

where  $x_{b \setminus j}$  collects all  $x_b$  with the exception of  $x_j$ . Here,  $\mu_{jb}(x_j)$  represents the message from a variable node  $j \in \mathcal{V}$  to a neighboring factor node  $b \in \mathcal{F}(j)$ , and the reverse for  $\mu_{bj}(x_j)$ . These messages are illustrated in Figure 1. The posterior belief can then be expressed as

$$q_j(x_j) = \frac{1}{Z_j} \mu_{jb}(x_j) \mu_{bj}(x_j), \tag{2.3}$$

with  $Z_j = \int \mu_{jb}(x_j) \mu_{bj}(x_j) dx_j$  a normalizing constant.

In practice, for numerical stability, messages are often renormalized after computation. Furthermore, messages are usually scheduled for computation and are often referred to by their position in the schedule instead of their location in the graph. We will use a similar notation in section 4. (See Bishop, 2006, for a more detailed introduction to (approximate) inference on bipartite graphs.)

**2.2 Bethe Free Energy Interpretation.** The Bethe free energy for a factorized model of the form of equation 2.1 is defined as

$$F[q] = \sum_{a \in \mathcal{F}} U_a[q_a] - \sum_{a \in \mathcal{F}} H[q_a] + (d_i - 1) \sum_{i \in \mathcal{V}} H[q_i], \tag{2.4}$$

where  $d_i$  represents the degree of variable  $x_i$ . Here  $U_a[q_a] = - \int q_a(x_a) \log f_a(x_a) dx_a$  denotes the average energy for factor  $f_a$ , and  $H[q_a] = - \int q_a(x_a)$

$\int \log q_a(x_a) dx_a$  denotes the entropy. The Bethe free energy is optimized with imposed normalization and marginalization constraints:

$$\int q_a(x_a) dx_{a \setminus j} = q_j(x_j), \forall a \in \mathcal{F}, \forall j \in \mathcal{V}(a), \quad (2.5a)$$

$$\int q_a(x_a) dx_a = 1, \forall a \in \mathcal{F}, \quad (2.5b)$$

$$\int q_i(x_i) dx_i = 1, \forall i \in \mathcal{V}, \quad (2.5c)$$

such that the  $q_a$  and  $q_i$  represent (approximate) posterior probability distributions (beliefs).

**2.3 Free Energy Minimization for Active Inference.** Active inference usually defines dynamic models that specialize variables into parameters, states, observation, and control sequences for past and future times. Free energy minimization for ActInf is then presented as a dual objective, where minimization of free energy for a model of past variables accounts for state and parameter estimation (perception), and free energy minimization of free energy for a model of future variables accounts for policy planning (Baltieri & Buckley, 2018; van de Laar, Özçelikkale, & Wymeersch, 2019).

In this letter, we assume that the current state is observed and that model parameters are given. Therefore, this letter concerns only inference for policy planning. Extensions for perception are, however, straightforward. Chance constraints only affect inference for planning, and therefore standard techniques for state estimation and parameter learning can be employed (van de Laar & de Vries, 2019).

Furthermore, this letter employs the Bethe free energy (BFE) formulation, equation 2.4 for policy planning (Schwöbel et al., 2018; van de Laar & de Vries, 2019) instead of the more traditional expected free energy (EFE) (Friston et al., 2015). The BFE is known to lack the epistemic qualities of the EFE (Schwöbel et al., 2018), which can be compensated for by introducing an additional mutual information term between the states and the observations to the BFE objective (Parr & Friston, 2019). The benefit of the uncompensated BFE, however, is that traditional message passing algorithms, including (loopy) belief propagation, variational message passing, expectation propagation, and generalized belief propagation algorithms, can all be derived as fixed-point equations of the variational free energy by the use of variational calculus (see Yedidia, Freeman, & Weiss, 2000; Heskes, 2003; Yedidia et al., 2005; Dauwels, 2007; Zhang et al., 2017).

**2.4 Chance Constraints.** A chance constraint imposes that the probability mass of a belief  $q_j(x_j)$ ,  $j \in \mathcal{V}$  outside of a “safe” region  $\mathcal{S}_j \subset \mathcal{X}_j$  cannot

exceed a preset threshold  $\epsilon \in [0, 1]$ . Formally, a chance constraint imposes the inequality

$$\begin{aligned} 1 - \epsilon &\leq \int_{\mathcal{S}_j} q_j(x_j) dx_j \\ &= \int_{\mathcal{X}_j} q_j(x_j) g_j(x_j) dx_j, \end{aligned} \quad (2.6)$$

with

$$g_j(x_j) = \begin{cases} 1 & \text{if } x_j \in \mathcal{S}_j \\ 0 & \text{otherwise.} \end{cases}$$

Our problem statement then becomes two-fold

1. Find the stationary points of the Bethe free energy, equation 2.4 under the normalization and marginalization constraints of equation 2.5 and chance constraints of the form 2.6 (theorem 1);
2. Interpret the retrieval of stationary points of the chance-constrained Bethe free energy as message passing on a factor graph (theorem 2).

The simulations of section 4 further specialize the model variables into state, observation, and control sequences and demonstrate the added value of chance constraints in an ActInf setting. Crucially, with an interpretation of chance constraints in terms of message passing on a factor graph, chance constraints can be readily applied to any factorized model. Formulating chance constraints as a click-on module for approximate inference then greatly improves the application range of chance constraints.

### 3 Chance-Constrained Message Passing

---

In this section, we formulate the method of chance-constrained message passing. We identify the stationary points of the chance-constrained Bethe free energy and interpret the result in terms of message passing on a factor graph. We work toward a practical message-passing update rule for chance-constrained variables, as summarized in algorithm 1. A brief introduction to variational calculus is available in appendix A. Proofs can be found in appendix B.

**3.1 Stationary Points.** From the Bethe free energy, equation 2.4, and the constraints of equations 2.5 and 2.6, we can construct the Lagrangian



---

**Algorithm 1** Chance-Constrained Message Passing with Gaussian Approximation.

---

Given a gaussian inbound message  $\mu_{jg}(x_j)$

Compute the uncorrected belief  $q_j^{(0)}(x_j)$  through equation 3.5a

Compute the safe mass  $\Phi_j^{(0)}$  through equation 3.5b

Initialize the approximated belief  $\tilde{q}_j^{(0)}(x_j) = q_j^{(0)}(x_j)$

Initialize the iteration counter  $n = 0$

**while**  $\epsilon + \delta < 1 - \Phi_j^{(n)}$  **do**

*% Chance constraint is violated with some tolerance  $\delta$*

Increase the counter  $n \leftarrow n + 1$

Compute the corrected belief  $q_j^{(n)}(x_j)$  through equation 3.9

Approximate  $\tilde{q}_j^{(n)}(x_j) \approx q_j^{(n)}(x_j)$  by gaussian moment matching

Compute  $\Phi_j^{(n)} = \int_{\mathcal{S}_j} \tilde{q}_j^{(n)}(x_j) dx_j$ , the safe mass of the approximated belief

**end while**

**return** The message  $\mu_{gj}(x_j) = \tilde{q}_j^{(n)}(x_j) / \mu_{jg}(x_j)$

---

$$\begin{aligned}
 L[q] = & F[q] + \sum_{i \in \mathcal{V}} \gamma_i \left[ \int q_i(x_i) dx_i - 1 \right] + \sum_{a \in \mathcal{F}} \gamma_a \left[ \int q_a(x_a) dx_a - 1 \right] \\
 & + \sum_{a \in \mathcal{F}} \sum_{i \in \mathcal{V}(a)} \int \zeta_{ia}(x_i) \left[ q_i(x_i) - \int q_a(x_a) dx_{a \setminus i} \right] dx_i \\
 & + \sum_{i \in \mathcal{V}} \eta_i \left[ \int q_i(x_i) g_i(x_i) dx_i - (1 - \epsilon) \right], \tag{3.1}
 \end{aligned}$$

where the Lagrange multipliers  $\gamma, \zeta, \eta$  enforce the constraints of equations 2.5 and 2.6.

Under strong duality, for the inequality constraint in equation 2.6, we have the complementary slackness condition (Boyd & Vandenberghe, 2004). This condition states that for optimality, we have  $\eta_i \left[ \int q_i(x_i) g_i(x_i) dx_i - (1 - \epsilon) \right] = 0$ . Therefore, either  $\eta_i > 0$ , which implies that the chance constraint of equation 2.6 holds with equality (active), or  $\eta_i = 0$ , which implies that the chance constraint may hold without equality (inactive). In other

words, the complementary slackness condition requires us to consider two scenarios: equation 2.6 holds with equality for  $\eta_i > 0$  and equation 2.6 is satisfied under  $\eta_i = 0$ . Hence, if  $\eta_i > 0$ , the chance constraint is activated and enforced with equality.

In lemmas 1 and 2 we express the stationary points of  $L[q]$  in terms of the beliefs. The proofs are presented in sections B.1 and B.2.

**Lemma 1.** *Stationary points of equation 3.1 as a functional of  $q_b, b \in \mathcal{F}$ , are of the form*

$$q_b^*(x_b) = \frac{1}{Z_b} f_b(x_b) \prod_{i \in \mathcal{V}(b)} \mu_{ib}(x_i), \tag{3.2}$$

with

$$Z_b = \int f_b(x_b) \prod_{i \in \mathcal{V}(b)} \mu_{ib}(x_i) dx_b$$

a normalizing constant.

**Proof.** See section B.1. □

Note that the  $\mu_{ib}$  have not yet been identified or interpreted as messages. We explicitly make this connection in section 3.3.

**Lemma 2.** *Stationary points of equation 3.1 as a functional of  $q_j, j \in \mathcal{V}$ , are of the form*

$$q_j^*(x_j; \eta_j) = \frac{1}{Z_j(\eta_j)} \exp(-\eta_j g_j(x_j)) \prod_{a \in \mathcal{F}(j)} \mu_{aj}(x_j), \tag{3.3}$$

with

$$Z_j(\eta_j) = \int \exp(-\eta_j g_j(x_j)) \prod_{a \in \mathcal{F}(j)} \mu_{aj}(x_j) dx_j$$

a normalizer that still depends on  $\eta_j$ .

**Proof.** See section B.2. □

Note that, in contrast to equation 2.3, this result incorporates an additional exponential term for  $\eta_j$ . We identify this multiplier in section 3.2. However, we already know that when the chance constraint for  $j$  is inactive,  $\eta_j = 0$  as a consequence of the complementary slackness condition. In this case, equation 3.3 reduces to equation 2.3.

**3.2 Active Chance Constraint.** In this section, we identify the stationary points under active chance constraint. The result is stated in theorem 1.

**Theorem 1.** *Under active chance constraint, stationary points of equation 3.1 as a functional of  $q_j$ ,  $j \in \mathcal{V}$  are of the form*

$$q_j^*(x_j; \eta_j = \eta_j^*) = \begin{cases} \frac{1-\epsilon}{\Phi_j^{(0)}} q_j^{(0)}(x_j) & \text{if } x_j \in \mathcal{S}_j \\ \frac{\epsilon}{1-\Phi_j^{(0)}} q_j^{(0)}(x_j) & \text{otherwise,} \end{cases} \quad (3.4)$$

with

$$q_j^{(0)}(x_j) = q_j^*(x_j; \eta_j = 0), \quad (3.5a)$$

$$\Phi_j^{(0)} = \int_{\mathcal{S}_j} q_j^{(0)}(x_j) dx_j, \quad (3.5b)$$

$$\eta_j^* = \log(\epsilon \Phi_j^{(0)}) - \log(1 - \epsilon) - \log(1 - \Phi_j^{(0)}). \quad (3.5c)$$

**Proof.** See section B.3. □

This remarkable result tells us that the corrected belief  $q_j^*(x_j; \eta_j = \eta_j^*)$  is obtained by scaling the probability mass of the uncorrected belief  $q_j^{(0)}(x_j)$  over the respective safe and unsafe regions. This defines the corrected belief as a mixture of truncated beliefs. The optimal scaling of equation 3.4 ensures that the overflow is equal to  $\epsilon$ .

The complementary slackness condition ensures that the chance constraint is only enforced if the probability mass of the unconstrained belief overflows the “safe” region  $\mathcal{S}_j$  by more than  $\epsilon$ ; that is, the uncorrected belief is “unsafe” when

$$\epsilon < 1 - \Phi_j^{(0)}, \quad (3.6)$$

where we refer to  $\Phi_j^{(0)}$  as the safe mass.

If equation 3.6 is satisfied, then the posterior density  $q_j^{(0)}(x_j)$  is corrected according to equation 3.4, which “pushes” the probability mass (just) back inside the safe region.

**3.3 Chance-Constrained Message Passing.** In this section, we show that chance constraints, equation 3.4, can be interpreted as auxiliary factor nodes (with a specific node-function) and can be enforced by belief propagation in an augmented graph.

**Theorem 2.** *Given a bipartite graph  $\mathcal{G} = (\mathcal{F}, \mathcal{V}, \mathcal{E})$  with a variable node  $j \in \mathcal{V}$ , and an associated Bethe free energy, equation 2.4, with a chance constraint,*

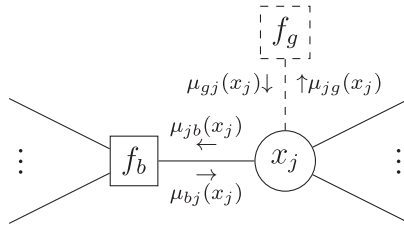


Figure 2: Bipartite graph around a chance-constrained variable  $x_j$ , with indicated auxiliary factor  $f_g$  (dashed square) and messages. Ellipses represent the continued model by an arbitrary (possibly zero) number of connected edges.

equation 2.6, on the belief  $q_j(x_j)$ . Then stationary points of equation 3.1 can be obtained by belief propagation on an augmented graph  $\mathcal{G}' = (\mathcal{F}', \mathcal{V}, \mathcal{E}')$ , where

$$\mathcal{F}' = \mathcal{F} \cup g, \tag{3.7a}$$

$$\mathcal{E}' = \mathcal{E} \cup (j, g), \tag{3.7b}$$

and auxiliary node function

$$f_g(x_j) = \begin{cases} \frac{1-\epsilon}{\Phi_j^{(0)}} & \text{if } x_j \in \mathcal{S}_j \\ \frac{\epsilon}{1-\Phi_j^{(0)}} & \text{otherwise.} \end{cases} \tag{3.8}$$

**Proof.** See section B.4. □

Theorem 2 shows that chance-constrained message passing can be seamlessly incorporated within the belief propagation framework. Chance constraints simply enter the model definition as auxiliary factors, whose factor function depends on the incoming message (see Figure 2). Because uncorrected belief, equation 3.5a, is being represented by the (renormalized) incoming message  $\mu_{jg}(x_j)$ , this allows for a modular application of chance constraints by augmenting the original graphical model with auxiliary nodes.

**3.4 Gaussian Approximation.** Since the message  $\mu_{gj}(x_j)$  introduces discontinuities, the computations for dependent messages may grow prohibitively complex. For efficient computations, it can be helpful to make a gaussian approximation  $\tilde{q}_j(x_j)$  to the corrected belief  $q_j^*(x_j; \eta_j = \eta_j^*)$ , for example, by moment matching. The resulting (approximate) message then follows from

$$\mu_{gj}(x_j) = \tilde{q}_j^{(n)}(x_j) / \mu_{jg}(x_j).$$

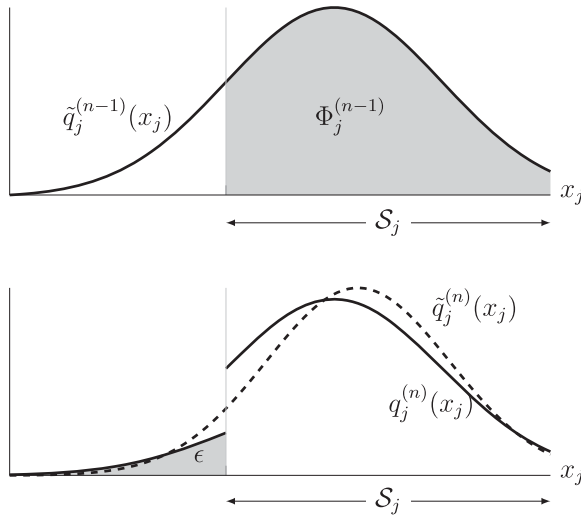


Figure 3: Example of beliefs as computed by algorithm 1. The top panel evaluates the probability mass within the “safe” zone. The bottom panel applies the correction (solid curve) and approximates the corrected belief by gaussian moment matching (dashed curve).

If the message  $\mu_{jg}(x_j)$  is also gaussian, this computation is easily performed by subtracting the canonical statistics. This procedure then resembles the expectation propagation algorithm (Minka, 2001; Cox & de Vries, 2018). Interestingly, this algorithm can also be derived in terms of Bethe free energy optimization, where the marginalization constraints, equation 2.5a, are replaced by moment-matching constraints (Zhang et al., 2017). This makes the gaussian approximation consistent with the Lagrangian approach as presented in this letter.

The approximated belief  $\tilde{q}_j(x_j)$ , however, renders the chance constraint, equation 2.6, inexact. As a result, the approximated belief needs to be iteratively recorrected:

$$q_j^{(n)}(x_j) = \begin{cases} \frac{1-\epsilon}{\Phi_j^{(n-1)}} \tilde{q}_j^{(n-1)}(x_j) & \text{if } x_j \in \mathcal{S}_j \\ \frac{\epsilon}{1-\Phi_j^{(n-1)}} \tilde{q}_j^{(n-1)}(x_j) & \text{otherwise,} \end{cases} \tag{3.9}$$

where  $n$  denotes an iteration counter. This leads to the procedure summarized in algorithm 1 and depicted in Figure 3.

With this algorithm, we have derived a practical chance-constrained message update from the first principles. The message update can be readily

applied to any continuous variable that requires a chance constraint. Note, however, that when multiple chance constraints are imposed on the model, the message passing algorithm itself becomes an iterative procedure because of circular message dependencies. For example, a message incoming to an auxiliary node  $g$  might (indirectly) depend on a message that exits another auxiliary node  $h$ . In turn, this exiting message depends on the incoming message to  $h$ , which depends on the message exiting  $g$ , and so on. In order to break this circular message dependency, uninformative messages can be used to initialize the algorithm.

#### 4 Simulations

In this section, we simulate a drone that aims to elevate itself above a given height threshold with a preset probability, under the influence of a stochastic vertical wind. We define the drone elevation level over time by  $x = \{x_0, \dots, x_t, \dots, x_L\}$ ,  $x_t \in \mathbb{R}$ , and actions (ascension velocity)  $a = \{a_0, \dots, a_t, \dots, a_L\}$ ,  $a_t \in \mathbb{R}$ . A time-dependent  $m_{w,t}$  defines the expected wind velocity that acts on the agent. The discrete-time stochastic system is defined as

$$w_t \sim \mathcal{N}(m_{w,t}, v_w),$$

$$x_{t+1} = x_t + a_t + w_t,$$

where  $v_w$  defines the wind velocity variance.

We define an agent that directly observes its elevation level and has knowledge of the statistical system properties  $m_{w,t}$  and  $v_w$ . The agent models future states of the system with a fixed time horizon  $T$ . As a shorthand notation, we write the future (including current) states  $\bar{x}_t = \{x_t, \dots, x_{t+T}\}$  and control variables  $\bar{u}_t = \{u_t, \dots, u_{t+T-1}\}$ . For notational convenience, we drop the  $t$  subscript from these collections. The agent model at time  $t$  is defined as

$$f_t(\bar{x}, \bar{u}) = \prod_{k=t}^{t+T-1} p_{x,k}(x_{k+1}|u_k, x_k)p_u(u_k), \quad (4.1)$$

with a respective state transition model and control prior,

$$p_{x,k}(x_{k+1}|u_k, x_k) = \mathcal{N}(x_{k+1}|x_k + u_k + m_{w,k}, v_w) \quad (4.2a)$$

$$p_u(u_k) = \mathcal{N}(u_k|0, \lambda^{-1}). \quad (4.2b)$$

We factorize and constrain the variational posterior distribution such that (van de Laar & de Vries, 2019),

$$q_t(\bar{x}_{\setminus t}, \bar{u}) = q_t(\bar{x}_{\setminus t}) \prod_{k=t}^{t+T-1} \delta(u_k - a_k), \tag{4.3}$$

where  $\bar{x}_{\setminus t}$  indicates the collection of latent states (the state sequence  $\bar{x}$  without the observed current state  $x_t$ ). The goal of the agent controller then becomes to find the policy  $\pi_t = \{a_t, \dots, a_{t+T-1}\}$  that minimizes the Bethe free energy,

$$F[q_t; x_t, \pi_t] = \int \dots \int q_t(\bar{x}_{\setminus t}, \bar{u}) \log \frac{q_t(\bar{x}_{\setminus t}, \bar{u})}{f_t(\bar{x}, \bar{u})} d\bar{x}_{\setminus t} d\bar{u}, \tag{4.4}$$

under the normalization and marginalization constraints of equation 2.5 and chance constraints

$$1 - \epsilon \leq \int_S q_{x,k}(x_k) dx_k, \forall k \in \{t + 1, \dots, t + T\},$$

where the safe region  $S = (1, \infty)$  and violation probability  $\epsilon$  are identical for all future state variables.

**4.1 Graphical Model and Schedule.** As detailed in section 3, Bethe free energy minimization under chance constraints can be performed by message passing on an augmented model. The graphical representation of the augmented model is depicted in Figure 4.

The schedule comprises a forward-backward scheme, as illustrated in Figure 5. Four message updates in Figure 5 are of particular interest. First, since equation 4.3 constrains the belief over controls to a point mass, it follows that

$$\mu_{\textcircled{2}}^{(i)}(u_k) = \delta(u_k - a_k^{(i-1)}),$$

where  $i$  counts the number of schedule (forward-backward) iterations. The schedule is initialized with  $a_k^{(0)} = 0$  for all  $k \geq t$ . Second,  $\mu_{\textcircled{B}}^{(i)}(x_{k+1})$  takes on the role of  $\mu_{jg}(x_j)$  in algorithm 1. Because the noise in the model is gaussian, this message will be an (unnormalized) gaussian as well. Therefore, by application of algorithm 1, the third message of interest,  $\mu_{\textcircled{6}}^{(i)}(x_{k+1})$  is computed. For the initial forward pass,  $\mu_{\textcircled{B}}^{(0)}(x_{k+1}) = 1$  is considered uninformative. Fourth,  $\mu_{\textcircled{P}}^{(i)}(u_k)$  carries information upward to the control variables. Because the variational posterior is chosen to factorize between the state and control sequence, equation 4.3, the  $\mu_{\textcircled{B}}^{(i)}(u_k)$  message is computed by a variational update rule as detailed in Winn and Bishop (2005) and Dauwels (2007).

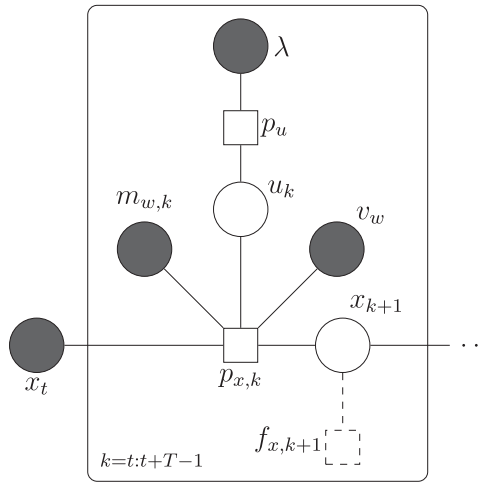


Figure 4: Augmented graphical representation of the agent model, equation 4.1. Circles and squares indicate variable and factor nodes respectively. Auxiliary factor nodes, equation 3.8, are dashed, and filled circles indicate observed variables or fixed parameters. Ellipses indicate a continuation of the framed section until the look-ahead time horizon.

The action for the next iteration then follows from

$$q_{u,k}^{(i)}(u_k) \propto \mu_{\text{Ⓟ}}^{(i)}(u_k) \mu_{\text{Ⓞ}}^{(i)}(u_k),$$

$$a_k^{(i)} = \text{mode} q_{u,k}^{(i)}(u_k).$$

Iterating the schedule then corresponds to an expectation-maximization scheme. The expectation step of this scheme computes the  $\mu_{\text{Ⓟ}}^{(i)}(u_k)$  message from the actions  $a_k^{(i-1)}$ . The maximization step then chooses the updated actions  $a_k^{(i)}$  as the current maximum a posteriori (MAP) estimate of  $u_k$ . The schedule is iterated until the policy converges.

Message passing simulations<sup>1</sup> are performed with the ForneyLab probabilistic programming toolbox (Cox, van de Laar, & de Vries, 2019), version 0.11.3.

**4.2 Control Law.** Note that the Bethe free energy of equation 4.4 is still a function of the observed current elevation  $x_t$ . We can then evaluate the optimal action  $a_t$  as a function of the current elevation  $x_t$  (the control law)

<sup>1</sup>Source code for the simulations is available for download at [http://biaslab.github.io/materials/cc\\_simulations.zip](http://biaslab.github.io/materials/cc_simulations.zip).



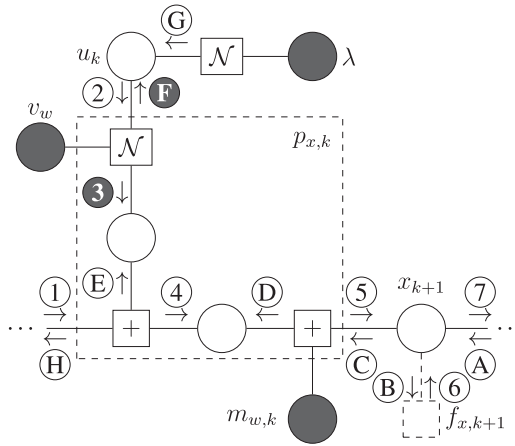


Figure 5: Augmented agent model, equation 4.1, with  $p_{x,k}$  expanded according to equation 4.2 (dashed rectangle), and indicated forward (numbers) and backward (letters) message passing schedules for optimization of equation 4.4. Circle and square nodes indicate variable- and factor-nodes respectively. Dark nodes indicate observed variables or fixed parameters, and auxiliary factor nodes, equation 3.8, are dashed. Ellipses indicate a continuation of the model. Dark messages are computed by the variational update rule. (See Winn & Bishop, 2005; Dauwels, 2007.)

for a given wind profile, chance constraint, and model parameters. In order to gain an intuition for controller behavior, we fix  $m_{w,t} = 0$  for all  $t$ . We plot the control law in Figure 6, for varying values of the look-ahead horizon  $T$ , chance constraint threshold  $\epsilon$ , wind variance  $v_w$ , and control prior precision  $\lambda$ .

The top-left diagram shows that with growing look-ahead horizon  $T$ , the agent starts intervening at higher elevation. With this anticipatory effect, the agent prepares for events in the more distant future. The top-right diagram also shows that the agent intervenes at higher elevation with decreasing  $\epsilon$ . When violation of the constraint grows less desirable, the agent must intervene earlier in order to ensure that sufficient probability mass is present in the safe region. Also note that no further action is proposed beyond an intervention threshold. Once the agent is sufficiently elevated, no corrections are proposed until the agent wanders (or is forced) below the intervention threshold. The bottom-left figure shows a similar effect for growing wind velocity variance  $v_w$ . When the system grows more stochastic, chance constraint abidance is ensured by intervening at higher elevations. Finally, the bottom-right figure illustrates what happens when the chance constraint is combined with a gaussian prior constraint on control. Increasing the control prior precision  $\lambda$  penalizes immediate correction. For low precisions (low

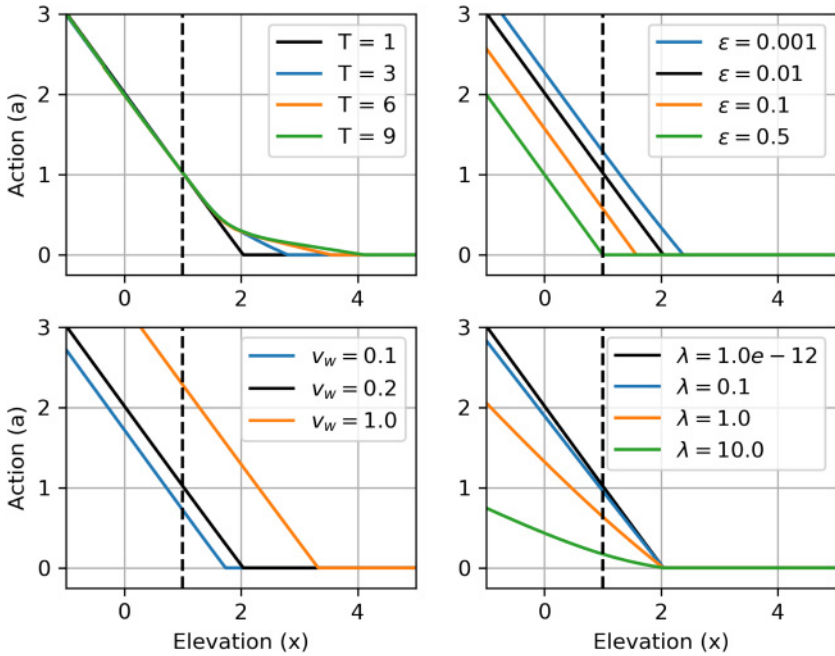


Figure 6: Slices of the control law for  $m_{w,t} = 0$ ,  $S = (1, \infty)$ , varied around reference setting  $T = 1$ ,  $\epsilon = 0.01$ ,  $v_w = 0.2$ ,  $\lambda = 10^{-12}$  (black curves). Dashed vertical lines indicate the minimal safe elevation.

penalty on control magnitude), the slope of the control law below the intervention threshold is equal to one, and compensation is immediate. Control grows more robust with growing precision, at the cost of prolonged chance constraint violation.

**4.3 Comparison against a Goal-Driven Agent.** In order to illustrate the difference in behavior between a chance- and a goal-driven ActInf agent, we compare the results of Figure 6 with an ActInf agent where the chance constraint is replaced by a goal prior. We use the graphical model definition of Figure 4 and define the auxiliary node function as a fixed prior  $f_{x,k+1}(x_{k+1}) = \mathcal{N}(x_{k+1}|m_x, \vartheta_x)$  for all  $t \leq k \leq t + T - 1$ . We choose  $m_x = 2$  and the variance  $\vartheta_x = 0.18478$  such that the overflow of the safe region  $1 - \int_S f_{x,k+1}(x_{k+1}) dx_{k+1} \approx 0.01$  resembles the situation for  $\epsilon = 0.01$ . The message passing schedule then follows the definition of Figure 5, where  $\mu_{\odot}^{(i)}(x_{k+1})$  is no longer computed by algorithm 1 and propagates the fixed goal prior instead. Figure 7 shows the resulting control law for  $m_{w,t} = 0$ ,  $T = 1$ ,  $v_w = 0.2$ , and varying  $\lambda$ .

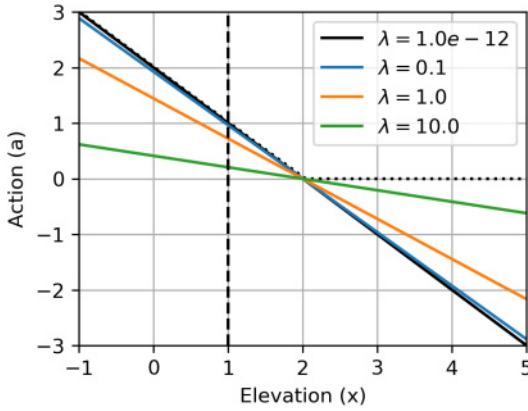


Figure 7: Slices of the control law for a goal-driven agent with  $m_{w,t} = 0, T = 1, m_x = 2, \vartheta_x = 0.18478, v_w = 0.2$  with varying  $\lambda$ . The dashed vertical line indicates the minimal safe elevation. The black dotted curve represents the reference result ( $\lambda = 10^{-12}$ ) for the chance-driven agent (Figure 6, black curves).

The results of Figure 7 show that the control for the goal-driven agent grows more robust with increasing  $\lambda$ , similar to the control law for the chance-driven agent (see Figure 6, bottom right). For the smallest  $\lambda$ , the control law for the prior-driven agent resembles the corresponding control law for the chance-driven agent (dotted curve) only for elevations  $x < 2$ . For elevations  $x > 2$ , the goal-driven agent proposes downward corrections, while the chance-driven agent proposes no corrections. This comparison illustrates how a chance-driven agent avoids unnecessary interventions.

**4.4 Simulation Results.** In this section, we study an active inference agent in interaction with a simulated environment. The action-perception loop is based on van de Laar and de Vries (2019) and consists of four steps at every time  $t$ :

1. *Observe* the current agent elevation.
2. *Infer* a policy from the current elevation and the future expected wind velocities by chance-constrained message passing.
3. *Act* by selecting the first (current) action from the inferred policy.
4. *Execute* the selected action in the system and advance the time index by one.

The results for 10,000 independent runs are plotted in Figure 8 for a chance-driven agent (left) and a goal-driven agent (right). The first row of diagrams plots the expected wind velocity over time, which is identical for each run. The sampled wind velocity trajectories  $w_t$  do vary per

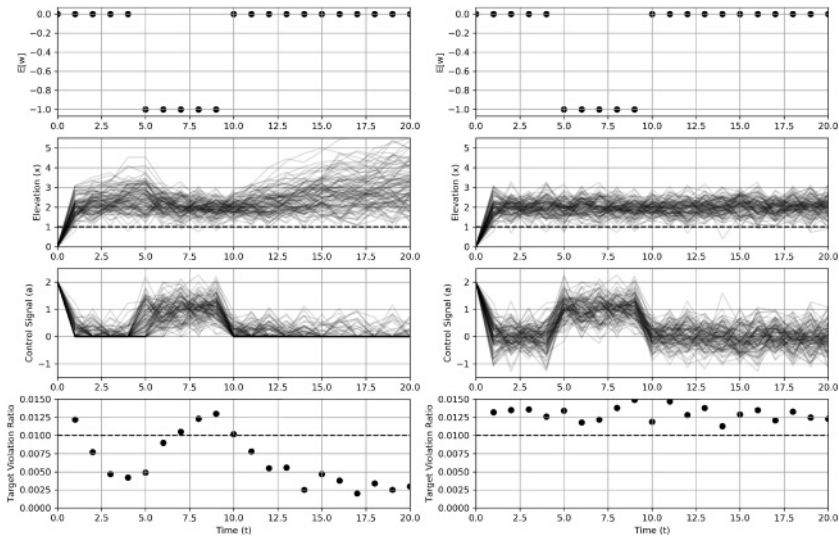


Figure 8: Results for 10,000 simulations with varying wind strength over time, and  $T = 1$ ,  $v_w = 0.2$ ,  $\lambda = 10^{-12}$ , for a chance-driven agent ( $\epsilon = 0.01$ , left), and a goal-driven agent ( $m_x = 2$ ,  $v_x = 0.18478$ , right).

run, under the influence of the wind velocity variance  $v_w$ . For  $5 \leq t < 10$ , a downward draft attempts to push the drone below the minimal safe elevation (dashed). The second row plots the drone elevation trajectory for a randomly selected subset of runs. Corresponding actions are plotted in the third row. The fourth row evaluates the relative number of runs that violate the safe zone over time.

It can be seen that both agents undertake corrective actions in order to compensate for the downward wind. However, while the chance-driven agent (left) only proposes upward corrections below the intervention threshold, the goal-driven agent (right) proposes additional downward corrections above the threshold. Furthermore, it can be seen that the maximal empirical violation for the chance-constrained agent mostly remains below the chance constraint target violation probability of  $\epsilon = 0.01$  (dashed), while the goal-driven agent systematically overshoots the target violation probability, that is, it violates the chance constraint. Compared to the chance-driven agent, the maximal empirical violations for the goal-driven agent are also larger. This effect can be explained in terms of the constrained beliefs. Namely, the chance-driven agent constrains the posterior beliefs, while the goal-driven agent imposes prior constraints on the model. Prior constraints may still be violated by the corresponding posterior beliefs, leading to more pronounced empirical violations.

## 5 Conclusion

---

In this letter, we formulated chance-constrained optimization of the Bethe free energy in terms of message passing on a factor graph. We showed that in the factor graph representation of the generative model, chance constraints can be imposed by auxiliary factors that force (a specified portion of) the probability mass of the chance-constrained beliefs inside a designated safe zone. Message passing on the augmented graph, with the auxiliary factor nodes included in the graph, then automatically balances the imposed chance constraints with additional (prior) constraints on the generative model. Chance constraints can thus be interpreted as modular click-on extensions to the generative model, similar to conventional factor-nodes (Loeliger et al., 2004), and can thus be used to complement message-passing formulations on generative neural models (Friston et al., 2017; van de Laar, Cox, Senoz, Bocharov, & de Vries, 2018).

However, because the analytical result for the chance-constrained update includes an inherent discontinuity, direct application of this rule may still lead to message updates that grow prohibitively complex. To remedy this, we proposed an algorithm that approximates the resulting message with a gaussian form. This algorithm offers a tractable formulation of chance-constrained message passing. The proposed message passing interpretation of chance constraints then vastly enhances the modularity and flexibility of chance-constrained inference and can accelerate the search for workable models (Blei, 2014).

We demonstrated chance-constrained message passing in the context of active inference. We compared the simulated behavior of a chance-driven agent with a goal-driven agent, where the chance constraints are replaced by traditional prior beliefs on future outcomes. The results illustrate how the goal-driven agent continually proposes corrections, whereas the chance-driven agent seizes interventions above a threshold. Chance-constrained ActInf may thus avoid unnecessary interventions and reduce the cost of control.

The results for the chance-driven agent showed that in the absence of additional prior constraints, the empirical chance constraint violation ratio mostly remains below the preset target violation probability. An added prior constraint on controls robustifies control at the cost of prolonged chance constraint violation. Chance-constrained active inference thus weights all imposed constraints on the generative model, allowing, for example, for a trade-off between robust control and empirical chance constraint violation.

## Appendix A: Calculus of Variations

---

The calculus of variations offers a principled method for optimizing functionals (a function of a function that returns a scalar). We follow (Engel &

Dreizler, 2013) and consider the impact of a variation in a function  $q(x)$ ,  $x \in \mathcal{X}$ , on a functional  $L[q]$ . We define an infinitesimal variation of  $q$  by

$$\delta q \triangleq \beta \phi,$$

where  $\beta \rightarrow 0$ , and  $\phi(x)$  is a continuous and differentiable “test” function.

The functional derivative  $\delta L/\delta q$  relates a variation in  $q$  to a change in  $L$ , by Parr (1980):

$$\left. \frac{dL[q + \beta \phi]}{d\beta} \right|_{\beta=0} = \int \frac{\delta L}{\delta q}(x) \phi(x) dx. \tag{A.1}$$

The procedure then becomes to apply the operations on the left-hand side to  $L$ , and bring it into the form of the right-hand side, which allows us to identify the functional derivative  $\delta L/\delta q$ . The stationary points  $q^*$  are then obtained by setting  $\delta L/\delta q \stackrel{!}{=} 0$  and solving for  $q$ .

## Appendix B: Proofs

---

**B.1 Proof of Lemma 1.** Application of equations A.1 to 3.1 as a functional of  $q_b$ , yields

$$\left. \frac{dL[q_b + \beta \phi_b]}{d\beta} \right|_{\beta=0} = \int \phi_b(x_b) \left[ \log \frac{q_b(x_b)}{f_b(x_b)} + 1 + \gamma_b - \sum_{i \in \mathcal{V}(b)} \zeta_{ib}(x_i) \right] dx_b.$$

Identifying the functional derivative  $\delta L[q_b]/\delta q_b$  and setting it to zero, we obtain

$$q_b^*(x_b) = f_b(x_b) \exp \left[ \sum_{i \in \mathcal{V}(b)} \zeta_{ib}(x_i) - \gamma_b - 1 \right]. \tag{B.1}$$

We now define  $\mu(x_j) = \exp \zeta(x_j)$  and apply the normalization constraint, which recovers equation 3.2. □

**B.2 Proof of Lemma 2.** Application of equation A.1 to 3.1 as a functional of  $q_j$ , yields

$$\begin{aligned} \left. \frac{dL[q_j + \beta \phi_j]}{d\beta} \right|_{\beta=0} &= \int \phi_j(x_j) \left[ -(d_j - 1) + \gamma_j \right. \\ &\quad \left. - (d_j - 1) \log q_j(x_j) + \sum_{a \in \mathcal{F}(j)} \zeta_{ja}(x_j) + \eta_j g_j(x_j) \right] dx_j. \end{aligned}$$

Identifying the functional derivative  $\delta L[q_j]/\delta q_j$  and setting it to zero yields

$$q_j^*(x_j) = \exp \left[ \frac{1}{d_j - 1} \left( 1 - d_j + \gamma_j + \sum_{a \in \mathcal{F}(j)} \zeta_{ja}(x_j) + \eta_j g_j(x_j) \right) \right], \quad (\text{B.2})$$

which is the first expression for  $q_j^*$ .

We can obtain a second expression for  $q_j^*$  by applying the marginalization constraint to the result of lemma 1. Substituting equation 3.2 in 2.5a,

$$\begin{aligned} q_j^*(x_j) &= \int q_b^*(x_b) dx_{b \setminus j} \\ &= \frac{1}{Z_b} \mu_{jb}(x_j) \int f_b(x_b) \overbrace{\prod_{\substack{i \in \mathcal{V}(b) \\ i \neq j}} \mu_{ib}(x_i)}^{\mu_{bj}(x_j)} dx_{b \setminus j}, \end{aligned} \quad (\text{B.3})$$

where we identified a new quantity  $\mu_{bj}(x_j)$  (note the reverse indexing).

Interestingly, the marginalization result of equation B.3 not only holds for the specific factor  $b$ , but for all factors that neighbor  $j$ . Therefore, by symmetry, we can iterate the relation of equation B.3 for all  $c \in \mathcal{F}(j)$ :

$$\prod_{c \in \mathcal{F}(j)} q_j^*(x_j) = \prod_{c \in \mathcal{F}(j)} \frac{1}{Z_c} \mu_{jc}(x_j) \mu_{cj}(x_j).$$

We choose to exclude  $b$  itself from the iteration on both sides and obtain

$$\prod_{\substack{c \in \mathcal{F}(j) \\ c \neq b}} q_j^*(x_j) = \prod_{\substack{c \in \mathcal{F}(j) \\ c \neq b}} \frac{1}{Z_c} \mu_{jc}(x_j) \mu_{cj}(x_j). \quad (\text{B.4})$$

We substitute equation B.2 in the left-hand side of equation B.4 and note that the product on the left-hand side now has  $d_j - 1$  terms, and that neither of these terms depends on  $c$ . This allows us to remove the  $d_j - 1$  terms from the exponent of equation B.2, which yields

$$\exp(1 - d_j + \gamma_j + \eta_j g_j(x_j)) \prod_{a \in \mathcal{F}(j)} \mu_{ja}(x_j) = \prod_{\substack{c \in \mathcal{F}(j) \\ c \neq b}} \frac{1}{Z_c} \mu_{jc}(x_j) \mu_{cj}(x_j).$$

Canceling duplicate terms and simplifying, we obtain an expression for  $\mu_{jb}$  as identified in equation B.3:

$$\mu_{jb}(x_j) \propto \exp(-\eta_j g_j(x_j)) \prod_{\substack{a \in \mathcal{F}(j) \\ a \neq b}} \mu_{aj}(x_j). \tag{B.5}$$

Finally, substituting equation B.5 back in B.3 and renormalizing, we recover equation 3.3.  $\square$

**B.3 Proof of Theorem 1.** We start from equation 3.3, and use the definitions of equation 3.5 to obtain

$$\int_{S_j} q_j^*(x_j; \eta_j) dx_j = \frac{\Phi_j^{(0)} \exp(-\eta_j)}{\Phi_j^{(0)} \exp(-\eta_j) - \Phi_j^{(0)} + 1},$$

which leads to

$$\exp(-\eta_j^*) = \frac{(1 - \epsilon)(1 - \Phi_j^{(0)})}{\epsilon \Phi_j^{(0)}}.$$

We have now identified the  $\eta_j$  multiplier. Substituting this result back in equation 3.3 recovers equation 3.4, which expresses the corrected belief in terms of the uncorrected belief.  $\square$

**B.4 Proof of Theorem 2.** From equation 3.3, we express the uncorrected belief in terms of the messages

$$q_j^{(0)}(x_j) = \frac{1}{Z_j^{(0)}} \prod_{a \in \mathcal{F}(j)} \mu_{aj}(x_j), \tag{B.6}$$

with  $Z_j^{(0)} = Z_j(\eta_j = 0)$ .

We now construct the augmented graph  $\mathcal{G}' = (\mathcal{F}', \mathcal{V}, \mathcal{E}')$  according to equation 3.7, and define a message

$$\mu_{gj}(x_j) = f_g(x_j), \tag{B.7}$$

with  $f_g(x_j)$  as defined by equation 3.8.

Substituting equations B.6 and B.7 in 3.4 then yields the corrected belief in terms of the messages



$$q_j^*(x_j; \eta_j = \eta_j^*) = \frac{1}{Z_j^{(0)}} \mu_{g_j}(x_j) \mu_{jg}(x_j), \quad (\text{B.8})$$

with

$$\mu_{jg}(x_j) = \prod_{\substack{a \in \mathcal{F}'(j) \\ a \neq g}} \mu_{aj}(x_j). \quad (\text{B.9})$$

The results of equations B.7 to B.9 can be interpreted as belief propagation, equations 2.2 and 2.3, on the augmented graph  $\mathcal{G}'$ .  $\square$

## Acknowledgments

---

This work was supported, in part, by GN Hearing A/S and the Swedish Research Council (grants 2015-04011 and 2018-03701).

## References

---

- Baltieri, M., & Buckley, C. L. (2018). The modularity of action and perception revisited using control theory and active inference. In *Proceedings of the Artificial Life Conference* (pp. 121–128). Cambridge, MA: MIT Press.
- Baltieri, M., & Buckley, C. L. (2019). Active inference: Computational models of motor control without efference copy. In *Proceedings of the 2019 Conference on Cognitive Computational Neuroscience*. New York: ACM.
- Bishop, C. M. (2006). *Pattern recognition and machine learning*. New York: Springer.
- Blackmore, L., Ono, M., & Williams, B. C. (2011). Chance-constrained optimal path planning with obstacles. *IEEE Transactions on Robotics*, 27(6), 1080–1094.
- Blei, D. M. (2014). Build, compute, critique, repeat: Data analysis with latent variable models. *Annual Review of Statistics and Its Application*, 1, 203–232.
- Borrelli, F., Bemporad, A., & Morari, M. (2017). *Predictive control for linear and hybrid systems*. Cambridge: Cambridge University Press.
- Boyd, S., & Vandenberghe, L. (2004). *Convex optimization*. Cambridge: Cambridge University Press.
- Cox, M., & de Vries, B. (2018). Robust expectation propagation in factor graphs involving both continuous and binary variables. In *Proceedings of the 2018 26th European Signal Processing Conference* (pp. 2583–2587). Piscataway, NJ: IEEE.
- Cox, M., van de Laar, T. W., & de Vries, B. (2019). A factor graph approach to automated design of Bayesian signal processing algorithms. *International Journal of Approximate Reasoning*, 104, 185–204.
- Dauwels, J. (2007). On variational message passing on factor graphs. In *Proceedings of the IEEE International Symposium on Information Theory* (pp. 2546–2550). Piscataway, NJ: IEEE.
- de Vries, B., & Friston, K. J. (2017). A factor graph description of deep temporal active inference. *Frontiers in Computational Neuroscience*, 11, 95.
- Engel, E., & Drezler, R. M. (2013). *Density functional theory*. New York: Springer.

- Fountas, Z., Sajid, N., Mediano, P. A., & Friston, K. (2020). *Deep active inference agents using Monte-Carlo methods*. arXiv:2006.04176.
- Friston, K., Rigoli, F., Ognibene, D., Mathys, C., Fitzgerald, T., & Pezzulo, G. (2015). Active inference and epistemic value. *Cognitive Neuroscience*, 6(4), 187–214.
- Friston, K. J. (2010). The free-energy principle: A unified brain theory? *Nature Reviews Neuroscience*, 11(2), 127–138.
- Friston, K. J., Kilner, J., & Harrison, L. (2006). A free energy principle for the brain. *Journal of Physiology, Paris*, 100(1–3), 70–87.
- Friston, K. J., Parr, T., & de Vries, B. (2017). The graphical brain: Belief propagation and active inference. *Network Neuroscience*, 1(4), 381–414.
- Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., . . . Bengio, Y. (2014). Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, & K. Q. Weinberger (Eds.), *Advances in neural information processing systems*, 27. Red Hook, NY: Curran.
- Heskes, T. (2003). Stable fixed points of loopy belief propagation are local minima of the Bethe free energy. In S. Becker, S. Thrun, & K. Obermayer (Eds.), *Advances in neural information processing systems*, 15 (pp. 359–366). Cambridge, MA: MIT Press.
- Hoffmann, C., & Rostalski, P. (2017). Linear optimal control on factor graphs: A message passing perspective. In *Proceedings of the 20th IFAC World Congress*. Amsterdam: Elsevier.
- Imohiosen, A., Watson, J., & Peters, J. (2020). Active inference or control as inference? A unifying view. In *Proceedings of the 1st International Workshop on Active Inference*. New York: Springer.
- Korl, S. (2005). *A factor graph approach to signal modelling, system identification and filtering*. ETH Zurich.
- Loeliger, H.-A., Dauwels, J., Koch, V. M., & Korl, S. (2004). Signal processing with factor graphs: examples. In *Proceedings of the First International Symposium on Control, Communications and Signal Processing, 2004* (pp. 571–574). Piscataway, NJ: IEEE.
- Markovic, D., Stojic, H., Schwoebel, S., & Kiebel, S. J. (2021). *An empirical evaluation of active inference in multi-armed bandits*. arXiv:2101.08699.
- Mesbah, A. (2016). Stochastic model predictive control: An overview and perspectives for future research. *IEEE Control Systems Magazine*, 36(6), 30–44.
- Millidge, B., Tschantz, A., Seth, A. K., & Buckley, C. L. (2020). On the relationship between active inference and control as inference. In *Proceedings of the First International Workshop on Active Inference*. New York: Springer.
- Minka, T. P. (2001). Expectation propagation for approximate Bayesian inference. In *Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence* (pp. 362–369). San Mateo, CA: Morgan Kaufmann.
- Parr, R. G. (1980). Density functional theory of atoms and molecules. In K. Fukui & A. Pullman (Eds.), *Horizons of quantum chemistry* (pp. 5–15). New York: Springer.
- Parr, T., & Friston, K. J. (2019). Generalised free energy and active inference. *Biological Cybernetics*, 113(5), 495–513.
- Pearl, J. (1982). Reverting Bayes on inference engines: A distributed hierarchical approach. In *Proceedings of the Second AAAI Conference on Artificial Intelligence* (pp. 133–136). Palo Alto, CA: AAAI.

- Ramstead, M. J. D., Badcock, P. B., & Friston, K. J. (2018). Answering Schrödinger's question: A free-energy formulation. *Physics of Life Reviews*, 24, 1–16.
- Recht, B. (2019). A tour of reinforcement learning: The view from continuous control. *Annual Review of Control, Robotics, and Autonomous Systems*, 2, 253–279.
- Sajid, N., Ball, P. J., Parr, T., & Friston, K. J. (2021). Active inference: Demystified and compared. *Neural Computation*, 33(3), 674–712.
- Sallans, B., & Hinton, G. E. (2001). Using free energies to represent Q-Values in a multi-agent reinforcement learning task. In T. Leen, T. Dietterich, & V. Tresp (Eds.), *Advances in neural information processing systems*, 13 (pp. 1075–1081). Cambridge, MA: MIT Press.
- Schwöbel, S., Kiebel, S., & Markovic, D. (2018). Active inference, belief propagation, and the Bethe approximation. *Neural Computation*, 30(9), 2530–2567.
- Tschantz, A., Millidge, B., Seth, A. K., & Buckley, C. L. (2020). *Reinforcement learning through active inference*. arXiv:2002.12636.
- Ueltzhöffer, K. (2018). Deep active inference. *Biological Cybernetics*, 112(6), 547–573.
- van de Laar, T. W. (2019). *Automated design of Bayesian signal processing algorithms*. Eindhoven University of Technology.
- van de Laar, T. W., Cox, M., Senoz, I., Bocharov, I., & de Vries, B. (2018). Forney-lab: A toolbox for biologically plausible free energy minimization in dynamic neural models. In *Proceedings of the Conference on Complex Systems*. New York: Springer.
- van de Laar, T. W., & de Vries, B. (2019). Simulating active inference processes by message passing. *Frontiers in Robotics and AI*, 6, 20.
- van de Laar, T. W., Özçelikkale, A., & Wymeersch, H. (2019). *Application of the free energy principle to estimation and control*. arXiv:1910.09823.
- Winn, J., & Bishop, C. M. (2005). Variational message passing. *Journal of Machine Learning Research*, 6, 661–694.
- Yedidia, J. S., Freeman, W. T., & Weiss, Y. (2000). Generalized belief propagation. In T. Leen, T. Dietterich, & V. Tresp (Eds.), *Advances in neural information processing systems*, 13 (pp. 689–695). Cambridge, MA: MIT Press.
- Yedidia, J. S., Freeman, W., & Weiss, Y. (2005). Constructing free-energy approximations and generalized belief propagation algorithms. *IEEE Transactions on Information Theory*, 51(7), 2282–2312.
- Zhang, D., Wang, W., Fettweis, G., & Gao, X. (2017). *Unifying message passing algorithms under the framework of constrained Bethe free energy minimization*. arXiv:1703.10932.

---

Received February 17, 2021; accepted May 3, 2021.