Sampling-Tree Model: Efficient Implementation of Distributed Bayesian Inference in Neural Networks

Zhaofei Yu, *Member, IEEE*, Feng Chen, *Member, IEEE*, and Jian K. Liu

***Abstract*—Experimental observations from neuroscience have suggested that the cognitive process of human brain is realized as probabilistic reasoning and further modeled as Bayesian inference. However, it remains unclear how Bayesian inference could be implemented by network of neurons in the brain. Here a novel implementation of neural circuit, named the sampling-tree model, is proposed to fulfill this aim. By using a deep tree structure to implement sampling with simple and stackable basic neural network motifs for any given Bayesian networks, one can perform local inference while guaranteeing the accuracy of global inference. We show that these task-independent motifs can be used in parallel for fast inference without intensive iteration and scale-limitation. As a result, this model utilizes the structure benefit of neuronal system, i.e., neuronal abundance and multihierarchy, to perform fast inference in an extendable way.**

***Index Terms*—Bayesian inference, importance sampling, neural network, probabilistic population coding (PPC), sampling-tree model (STM).**

I. Introduction

UNDERSTANDING how the brain works is one of the most challenging problems in the 21st century. Our brain can represent probability distribution [1]–[3]. The cognitive and perceptive process of the brain is a process of probabilistic reasoning, which has been indicated by a number of psychological and neuroscience experiments [4], [5]. From the macroscopic level, Bayesian models have shown their ability of explaining how the brain perceives the world and have been successfully used in various fields of brain science, such as perception [6]–[9], cognition [10]–[12], sensorimotor control [5], [13], [14], and decision making [15]–[18]. Nevertheless, from the microscopic perspective, it remains largely unknown how Bayesian inference is implemented by our neuronal systems, or more precisely, how can a network of spiking neurons implement inference algorithms of Bayesian models. Therefore, it is challenging, yet of great importance, to build the bridge between Bayesian inference models and possible implementations in a neural network. For one thing, it would help us understand the process of human cognition theoretically [3]. For another, recent advancements of neuromorphic chips can improve the computation power by utilizing neural circuits implementation of Bayesian inference [19]–[23].

According to recent studies, many types of neural networks (circuits) with different architectures have been proposed to perform inference of probabilistic graphical models, especially a Bayesian network. These neural networks differ in the way of expressing probability, which can be classified as the probability code, the log probability code, the population code, and the sampling-based code [24], [25]. Anastasio *et al.* [26] used the explicit probability code to express probabilities by assuming that the probabilities are proportional to the neuronal response in superior colliculus. In this way, the summation of probabilities can be calculated by summing the overall responses of neurons. The same way of coding was also used in [27]. In order to simplify the multiplication of probabilities, Rao [28], [29] proposed to use the log probability code and proved that the differential equations of recurrent neural networks are in coincidence with the inference equations of the hidden Markov model, in which the computation of sum-logs was used to approximate the computation of log-sum. Beck and Pouget [30] focused on this approximation problem and set up a precise equivalence relation from the first principle. Angela and Dayan [31] employed the same way of coding and built a hierarchy neural network to perform inference of posterior probabilities.

Another important way of coding is probabilistic population coding (PPC) [32], [33], which uses a population of neurons to encode a distribution, instead of probability values. Ma *et al.* [32] showed that cue integration can be implemented by linear combination of each population activity with PPC. The method was exploited thereafter by

Manuscript received April 2, 2019; revised June 12, 2019; accepted July 7, 2019. Date of publication July 10, 2019; date of current version September 9, 2020. This work was supported in part by the National Natural Science Foundation of China under Grant 61806011, Grant 61671266, and Grant 61836004; in part by the National Post-Doctoral Program for Innovative Talents under Grant BX20180005; in part by the China Post-Doctoral Science Foundation under Grant 2018M630036; in part by the International Talent Exchange Program of Beijing Municipal Commission of Science and Technology under Grant Z181100001018026; in part by the Royal Society Newton Advanced Fellowship under Grant NAF/R1/191082; and in part by the Tsinghua University Initiative Scientific Research Program under Grant 20161080084. *(Corresponding authors: Feng Chen; Jian K. Liu.)*

Z. Yu is with the National Engineering Laboratory for Video Technology, Department of Computer Science and Technology, Peking University, Beijing 100871, China, also with the Department of Automation, Center for Brain-Inspired Computing Research, Tsinghua University, Beijing 100084, China, and also with Peng Cheng Laboratory, Shenzhen 518055, China (e-mail: yuzf12@pku.edu.cn).

F. Chen is with the Department of Automation, Center for Brain-Inspired Computing Research, Tsinghua University, Beijing 100084, China, also with the Beijing Innovation Center for Future Chip, Beijing 100084, China, and also with the Beijing Key Laboratory of Security in Big Data Processing and Application, Beijing 100084, China (e-mail: chenfeng@tsinghua.edu.cn).

J. K. Liu is with the Centre for Systems Neuroscience, Department of Neuroscience, Psychology and Behaviour, University of Leicester, Leicester LE1 7HA, U.K., and also with Peng Cheng Laboratory, Shenzhen 518055, China (e-mail: jian.liu@leicester.ac.uk).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCDS.2019.2927808

Beck *et al.* [34] to realize Bayesian decision-making [15] and inference of marginalization. In addition, Ma and Rahmati [35] implemented causal inference with PPC. The above mentioned probabilistic codes can be summarized as the assumption that the physiological signals of neurons as a whole follow certain probability distribution. And yet there is another coding method, termed sampling-based coding, which treats neuronal spikes as samples from a particular probability distribution. Buesing *et al.* [36] and Pecevski *et al.* [37] proposed a method to perform inference of marginal probability based on Markov chain Monte Carlo (MCMC) as long as the network meets the neural computability condition (NCC). Shi and Griffiths [38] designed a neural network to implement hierarchical Bayesian inference by importance sampling, but it is limited to simple Bayesian models such as the chain model.

Most of the approaches described above consider how posterior probabilities are represented and optimized. There is a final body of work that deal directly with hierarchical Bayesian inference in the brain from a cognitive neurosciences viewpoint, which is called hierarchical predictive coding. This is a Bayesian filtering scheme that can be formally related to hierarchical extended Kalman filtering (and related to sampling approaches such as particle filtering). There is a large amount of anatomical and physiological evidence suggesting that the visual process uses some form of hierarchical predictive coding [39].

In summary, all these works focus on how a single neuron or a group of neurons implement probabilistic inference of probabilistic graphical models with a small number of nodes and edges. Just as concluded in [1], "Most studies in neuroscience have focused on problems with a small number of variables, all following simple distributions, for which an optimal solution can be easily derived... Real-life problems, however, are almost always far too complicated to allow for optimal behavior." Besides, as most of the previous studies take advantage of task-specific neural circuit, they are hard to be generalized to solve other inference problems [35]. It is worth considering how to build general-purpose neural networks for large-scale Bayesian models, and that is the goal of this paper. In order to achieve this, the neural network should resemble to the organization structure of the brain. Therefore, we propose four brain-inspired principles for designing of neural networks to implement Bayesian inference.

1) *Scalability:* The large number of neurons should be taken into account given that there are about 80 billion neurons in human brain, which brings powerful representation ability.
2) *Hierarchy:* The neural network has a hierarchical structure similar to human brain and it could extract information layer by layer.
3) *Locality:* A single neuron or a group of neurons should work in a simple style while complex functions could be achieved when they are connected together.
4) *Parallelizability:* The distributed neurons are organized to perform parallel computing simultaneously so that the inference is rapid enough for different tasks.

Based on aforementioned principles and our previous work of a sampling-based distributed inference algorithm [40], we propose a sampling-tree model (STM) as a neural network model for Bayesian inference. We characterize this model as STM because it is a probabilistic graphical model with hierarchical tree structure on the whole and enormous neurons representing samples at each node. In this model, the root node represents the problem we would like to infer, such as the inference of a stimulus, or the recognition of an object. The leaf nodes are the evidence we receive from the outside world. The branch nodes represent the intermediate variables.

In short, the main idea of the STM is to perform neural sampling on a deep tree-structured neural circuit. By taking full advantage of the tree structure, the global inference problem can be converted to the local inference problem. In consequence, we are able to design simple and repeatable basic neural network motifs to perform local reasoning while guaranteeing the accuracy of global reasoning. On the local level, importance sampling is introduced to conduct inference, which utilizes a massive number of neurons to sample in parallel so that the posterior probabilities can be calculated without iteration. This means that the STM takes the strategy of trading space for time and the inference process could be quite rapid. We also prove that the proposed model is able to approximate Bayesian inference with high accuracy. Experimental simulations, including integration of multicue information and object detection with compositional model, demonstrate that the STM is a general-purpose neural network, which can be used for distributed large-scale Bayesian inference.

To summarize, our contributions include the following aspects.

1) We propose a neural circuits model that can implement sample-based inference algorithm, and further implement fast and accurate inference of arbitrary Bayesian networks.
2) We prove that the particular independence assumptions of the inference algorithm can be effectively ignored.
3) We show that our proposed neural circuit can be used to solve practical cognitive problems, like integration of multicue information and object detection.
4) We give a functional explanation for neuronal abundance and multihierarchy of the brain from a computational perspective.

The rest of this paper is organized as follows. We first discuss the definition of the STM and show how to represent Bayesian models with the STM in Section II. Then, we show how to perform Bayesian inference with importance sampling from the algorithm level in Section III. Section IV gives some theoretical analysis of the proposed sampling-based inference algorithm. The detailed implementations of Bayesian inference with the STM from the neural circuits level are proposed in Section V. We show the experimental results in Section VI and conclude in Section VII. Part of this work was published as a short conference communication [40].

II. Definition of Sampling-Tree Model

In order to build a general-purpose neural network for large-scale Bayesian models, we propose the STM as shown in Fig. 1(a). From the macroscopic viewpoint, this model could

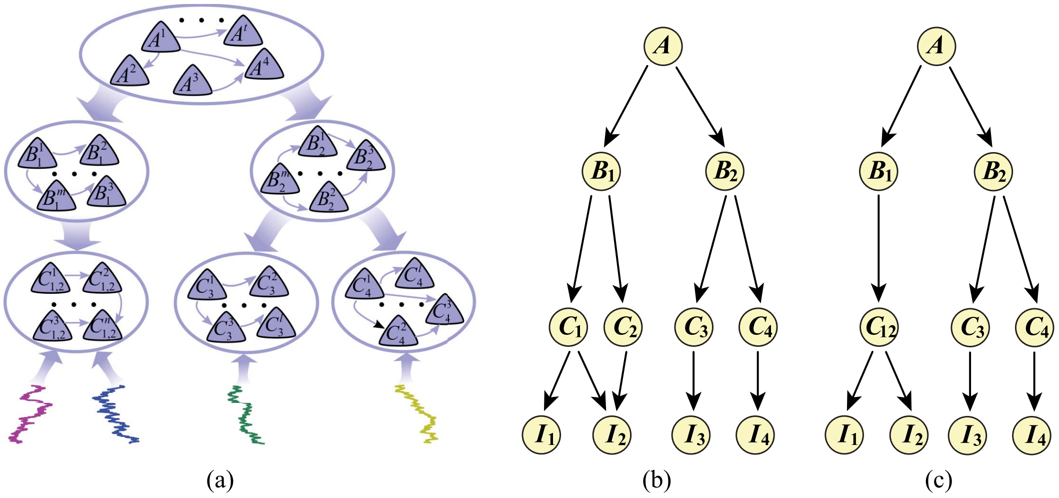


Fig. 1. STM. (a) Example of STM in neural network, where different evidence feeds into the different groups of neurons in a distributed way. Local computations are done by each group. (b) Nontree structured Bayesian model. (c) Tree-structured Bayesian model corresponding to the STM in (a). A nontree structured Bayesian model can be converted to a tree-structured Bayesian model by combining some variables, C_1 and C_2 here, together into one variable $C_{1,2}$.

be treated as a probabilistic graphical model with a hierarchical tree structure. From the neural level, each node includes a single neuron or a group of neurons representing samples and a number of connections between these neurons. In the STM, the root node represents the problem we want to infer, such as inference of outside stimuli or recognition of an object. The leaf nodes are the evidence we receive from the outside world. The branch nodes represent intermediate variables. Each neuron is viewed as a sample from a special distribution.[1] The connections between neurons are the basis of information transmission or probability calculation, which will be explained in the next sections. In summary, the STM we proposed has a hierarchical structure and includes large numbers of neurons, which is in accordance to the first two principles of brain-inspired neural network architecture, *scalability* and *hierarchy*.

The STM is able to represent tree-structured Bayesian inference because it is a hierarchical tree-structured model on the whole. The difficulty is how to represent nontree structured Bayesian models. Here we use the conclusion that by combining some variables together, one can convert a nontree structured Bayesian model into a tree-structured Bayesian model at the cost of greater state space [41, Ch. 10]. This means that in order to express all the states of a new variable, more neurons are needed than before. As long as there are enough neurons, the STM could represent any kind of Bayesian model.

Fig. 1(b) and (c) illustrates how to convert nontree structured Bayesian models into a tree-structured Bayesian model. Here the nontree structured model can be converted to a tree-structured Bayesian model by combing variables C_1 and C_2 to get a new variable $C_{1,2}$ [shown in Fig. 1(c)], and this new model is the same as the tree-structured model of the STM in Fig. 1(a), where a population of neurons are used to express a node. Consequently, the STM in Fig. 1(a) represents the nontree structured Bayesian models in Fig. 1(b). Supposing that the number of the states of variables C_1 and C_2 are both 10, the number of the states of variable $C_{1,2}$ will be 100. If each neuron represents a special state, then more neurons are needed to represent the combined variable $C_{1,2}$ than to represent variables C_1 and C_2. In fact, the Bayesian models used for real-life problems may include many nontree structures. As a result, the STM needs numerous neurons when representing these Bayesian models.

III. Bayesian Inference With Importance Sampling

In this section, we propose a sampling-based algorithm to perform Bayesian inference. We will explain the neural network architectures of STM that implement this algorithm in Section V. The Bayesian models discussed here are tree-structured Bayesian models. There are two reasons to study this kind of model. First, it is easy to perform inference of tree-structured Bayesian models [41]. Variational-based and sampling-based inference methods, like belief propagation (BP) [41], [42] and MCMC [43], [44], are able to perform accurate or nearly accurate inference with the benefit of tree structure. Second, the tree-structured models are capable of standing for many nontree structured models as an arbitrary Bayesian model could be converted to a tree-structured Bayesian model by combining some variables together [41].

Bayesian models for real-world problems are complex and the scale size can be very large. Although BP and MCMC can get accurate inference results in some Bayesian models, the existing neural networks implementing inference of these models with BP [45]–[47] or MCMC [36], [37] are very complicated. Each neuron or a group of neurons in these neural networks are commonly required to realize different and complex calculations, which violates the basic principle of neural system that a single neuron or a group of neurons should work in a simple style, whereas complex functions could be achieved when they are connected together. In addition, it takes considerable time for neural networks to converge to the inference result as they need multiple iterations. It is imperative

[1] As different neurons have different tuning curves, they can represent different states of a variable.

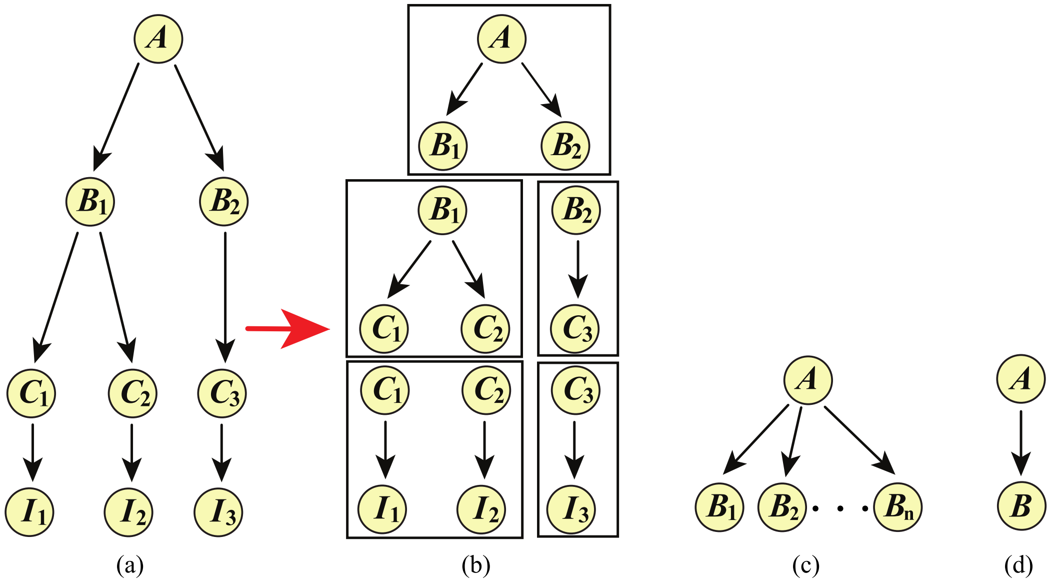


Fig. 2. Decomposition of tree-structured Bayesian network. (a) Example of tree-structured Bayesian network. (b) This Bayesian network is composed of basic network motifs. (c) Basic network motif in each box of (b) is a simple two-layer Bayesian network that consists of a parent node and several children nodes. (d) Special case of the basic network in (c).

to propose a new and fast inference algorithm and the corresponding neural circuits should and could be implemented by simple and basic networks. Thanks to the tree structure of Bayesian networks, global inference can be converted to local inference with network decomposition. The local inference problem is then performed by importance sampling, which takes advantage of massive numbers of neurons to sample in parallel. This means the STM can trade space for time so that inference would be quite rapid. Besides, this scheme of local inference guarantees that basic neural network of the STM is simple, which makes STM plausible for large-scale distributed computations.

A. Decomposition of Global Inference to Local Inference

The inference problem considered in this paper includes marginal inference and maximum a posterior (MAP) estimation. By marginal inference, we refers to computing the posterior of the root node being in each state given the state of the leaf nodes. Conversely, MAP estimation refers to finding the most probable state of the root node given the state of leaf nodes.

Specifically, we consider the tree-structured Bayesian network shown in Fig. 2(a), where A represents the root node, and I_1–I_3 denote the leaf nodes. The joint distribution defined on this Bayesian network has the form $P(A, B_1, B_2, C_1, C_2, C_3, I_1, I_2) = P(A)\ P(B_1|A)P(B_2|A)$ $P(C_1|B_1)P(C_2|B_1)P(C_3|B_2)P(I_1|C_1)\ P(I_2|C_2)P(I_3|C_3)$. If we have known the prior probability $P(A)$ and all the conditional probabilities defined on the right side of the equality defined above, the inference problem has the following two steps.

1) *Marginal Inference:* $P(A|I_1, I_2, I_3)$.
2) *MAP Estimation:* $\arg\max_A P(A|I_1, I_2, I_3)$.

As we can see, when performing marginal inference or MAP estimation of a tree-structured Bayesian network, the belief propagates from bottom to up. A direct idea is to decompose the network into simple and similar networks, then design an inference algorithm for each basic network. Each network could receive belief from all the children networks and at the same time pass its belief to the parent network. The similar structure in all the basic networks and the same inference algorithm guarantee that the whole neural network is composed of basic and repeatable neural network motifs. By analyzing the model in Fig. 2(a), we find that there is only one basic network, which consists of several children nodes and a parent node [shown in Fig. 2(b)–(d)]. If we can propose a rapid inference algorithm for the basic network and design a neural network to implement the algorithm, then the basic networks motifs can be combined to implement inference of the whole Bayesian network.

B. Inference of Tree-Structured Bayesian Models With Importance Sampling

In this paper, we conduct inference for the basic network motif with importance sampling, which is a method to estimate the value of some function by sampling from a simple distribution rather than the distribution of the interest [48], [49]. Actually importance sampling has been used to estimate the conditional expectation of some functions $f(x)$ given the variable y [38]

$$\begin{aligned} E(f(x)|y) &= \sum_x f(x)P(x|y) = \frac{\sum_x f(x)P(y|x)P(x)}{\sum_x P(y|x)P(x)} \\ &= \frac{E(f(x)P(y|x))_{P(x)}}{E(P(y|x))_{P(x)}} \approx \sum_{x^i} f\left(x^i\right)\frac{P\left(y|x^i\right)}{\sum_{x^i} P\left(y|x^i\right)} \\ & \quad x^i \sim P(x) \end{aligned} \tag{1}$$

where $x^i \sim P(x)$ denotes that x^i follows the distribution $P(x)$. Note that (1) converts the conditional expectation $E(f(x)|y)$ to the weighted combination of normalized conditional probabilities $[(P(y|x^i))/(\sum_{x^i} P(y|x^i))]$.

We generalize (1) to conduct inference of the basic Bayesian network in Fig. 2(c), where the problem is to compute $\sum_{B_1,B_2,\ldots,B_n} P(A|B_1, B_2, \ldots, B_n) \cdot P(B_1|I_1)P(B_2|I_2)\ldots P(B_n|I_n)$, and $I_1, I_2, \ldots, I_n$ represent evidence variables of $B_1, B_2, \ldots, B_n$, respectively [not shown in Fig. 2(c)]. One can drive the following equation with importance sampling:

$$\begin{aligned}
&\sum_{B_1,B_2,\ldots,B_n} P(A|B_1, B_2, \ldots, B_n)P(B_1|I_1)\cdots P(B_n|I_n)\\
&\approx \sum_{B_1,B_2,\ldots,B_n} P(A|B_1, B_2, \ldots, B_n)\\
&\quad\times P(B_1, B_2, \ldots, B_n|I_1, \ldots, I_n)\\
&\approx \sum_i P\big(A|B_1^i, B_2^i, \ldots, B_n^i\big)\\
&\quad\times \frac{P\big(I_1, I_2, \ldots, I_n|B_1^i, B_2^i, \ldots, B_n^i\big)}{\sum_i P\big(I_1, I_2, \ldots, I_n|B_1^i, B_2^i, \ldots, B_n^i\big)}\\
&= \sum_i P\big(A|B_1^i, B_2^i, \ldots, B_n^i\big)\\
&\quad\times \frac{P\big(I_1|B_1^i\big)P\big(I_2|B_2^i\big)\cdots P\big(I_n|B_n^i\big)}{\sum_i P\big(I_1|B_1^i\big)P\big(I_2|B_2^i\big)\cdots P\big(I_n|B_n^i\big)}\\
&B_1^i, B_2^i, \ldots, B_n^i \sim P(B_1, B_2, \ldots, B_n). \qquad (2)
\end{aligned}$$

Note that (2) is a function of variable A, and it can be further utilized when A is a child node of other nodes. Note that an approximation exists in (2), which is

$$\begin{aligned}
&P(B_1|I_1)P(B_2|I_2)\cdots P(B_n|I_n)\\
&\quad\approx P(B_1, B_2, \ldots, B_n|I_1, I_2, \ldots, I_n). \qquad (3)
\end{aligned}$$

This approximation can be understood like this. According to the total probability formula, $P(B_1, B_2, \ldots, B_n|I_1, I_2, \ldots, I_n)$ equals

$$\begin{aligned}
&P(B_1, B_2, \ldots, B_n|I_1, I_2, \ldots, I_n)\\
&\quad= P(B_1|I_1, I_2, \ldots, I_n)P(B_2|B_1, I_2, \ldots, I_n)\cdots\\
&\qquad P(B_n|B_1, B_2, \ldots, B_{n-1}, I_n). \qquad (4)
\end{aligned}$$

By comparing (3) and (4), we obtain

$$\begin{aligned}
&P(B_1|I_1)P(B_2|I_2)\cdots P(B_n|I_n)\\
&\quad\approx P(B_1|I_1, I_2, \ldots, I_n)P(B_2|B_1, I_2, \ldots, I_n)\cdots\\
&\qquad P(B_n|B_1, B_2, \ldots, B_{n-1}, I_n). \qquad (5)
\end{aligned}$$

Assumptions that satisfy (5) include a set equality of the following sort:

$$\begin{aligned}
P(B_1|I_1) &= P(B_1|I_1, I_2, \ldots, I_n)\\
P(B_2|I_2) &= P(B_2|B_1, I_2, \ldots, I_n)\\
&\cdots\\
P(B_n|I_n) &= P(B_n|B_1, B_2, \ldots, B_{n-1}, I_n). \qquad (6)
\end{aligned}$$

It implies that some conditional independence assumptions exist in (3) and (5), such as $B_1 \perp I_2, \ldots, I_n \mid I_1$, $B_2 \perp B_1, I_3, \ldots, I_n \mid I_2$, ..., $B_n \perp B_1, B_2, \ldots, B_{n-1} \mid I_n$. A special case of the network in Fig. 2(c) is that the parent node A has only one child node [shown in Fig. 2(d)] and (2) can be converted to

$$\begin{aligned}
\sum_B P(A|B)P(B|I) &\approx \sum_{B^i} P\big(A|B^i\big)\frac{P\big(I|B^i\big)}{\sum_{B^i} P\big(I|B^i\big)}\\
&\quad B^i \sim P(B). \qquad (7)
\end{aligned}$$

Note that there are no conditional independence assumptions in (7).

As arbitrary tree-structured Bayesian network could be divided into basic networks in Fig. 2(c) and (d), inference of tree-structured Bayesian network can be implemented by the composition of (2) and (7). Here we give an example to illustrate it. The inference problems in Fig. 2(a) are marginal inference $P(A|I_1, I_2, I_3)$ and MAP estimation $\arg\max_A P(A|I_1, I_2, I_3)$, among which marginal inference can be performed by (8), shown at the top of the next page.

Here $C_1^i, C_2^i \sim P(C_1, C_2)$, $C_3^j \sim P(C_3)$, $B_1^k, B_2^k \sim P(B_1, B_2)$, and $A^l \sim P(A)$. $I(A^l = a_t)$ is an indicator function, which equals to 1 only when $A^l = a_t$. Note that a_t is the possible state of the variable A and $t = 1, 2, \ldots, T$. Equation (8) includes some approximations

$$\begin{aligned}
P(C_1, C_2|I_1, I_2, I_3) &\approx P(C_1, C_2|I_1, I_2)\\
P(C_3|C_1, C_2, I_3) &\approx P(C_3|I_3)\\
P(B_1|C_1, C_2, C_3) &\approx P(B_1|C_1, C_2)\\
P(B_2|B_1, C_3) &\approx P(B_2|C_3)\\
P\Big(B_1, B_2|C_1^i, C_2^i, C_3^j\Big) &\approx P\big(B_1|C_1^i, C_2^i\big)P\Big(B_2|C_3^j\Big) \qquad (9)
\end{aligned}$$

which implies that (8) includes some conditional independence assumptions, that are $C_1, C_2 \perp I_3|I_1, I_2$, $C_1, C_2 \perp C_3|I_3$, $B_1 \perp C_3|C_1, C_2$, $B_1 \perp B_2|C_3$, $B_1 \perp C_3^j|C_1^i, C_2^i$, and $B_1 \perp B_2|C_3^j$.

In addition, MAP estimation is to choose the state that maximizes the posterior probability, which can be implemented easily after we have known the posterior distribution $P(A|I_1, I_2, I_3)$.

C. Generation of Samples From Prior Distributions With Importance Sampling

The precondition of the proposed algorithm is that the samples are generated from some special distributions, like prior distributions, however, not all of these special distributions are known. For example, considering the inference problem in Fig. 2(a), we suppose that the samples are generated from the distributions $P(C_1, C_2)$, $P(C_3)$, $P(B_1, B_2)$, and $P(A)$ in (8) while we only know the prior distribution $P(A)$. Therefore, one should propose an algorithm to sample from these special distributions and it should be able to be implemented by the STM. Interestingly, we find that importance sampling could solve this problem

$$\begin{aligned}
P(B_1, B_2) &= \sum_A P(A, B_1, B_2) = \sum_A P(A)P(B_1, B_2|A)\\
&= \frac{1}{L}\sum_{l=1}^{L} P\Big(B_1, B_2|A^l\Big), A^l \sim P(A). \qquad (10)
\end{aligned}$$

Here A^l follows the distribution $P(A)$. Then the probabilities $P(C_1, C_2)$ and $P(C_3)$ could be computed based on $P(B_1, B_2)$.

$$\begin{aligned}
&P(A = a_t|I_1, I_2, I_3)\\
&\quad= \sum_{A,B_1,B_2,C_1,C_2,C_3} I(A = a_t)P(A, B_1, B_2, C_1, C_2, C_3|I_1, I_2, I_3)\\
&\quad= \sum_{A,B_1,B_2,C_1,C_2,C_3} I(A = a_t)P(C_1, C_2, C_3|I_1, I_2, I_3)P(B_1, B_2|C_1, C_2, C_3)P(A|B_1, B_2)\\
&\quad= \sum_{A,B_1,B_2,C_1,C_2,C_3} I(A = a_t)P(C_1, C_2|I_1, I_2, I_3)P(C_3|C_1, C_2, I_3)P(B_1, B_2|C_1, C_2, C_3)P(A|B_1, B_2)\\
&\quad\approx \sum_{A,B_1,B_2,C_1,C_2,C_3} I(A = a_t)P(C_1, C_2|I_1, I_2)P(C_3|I_3)P(B_1, B_2|C_1, C_2, C_3)P(A|B_1, B_2)\\
&\quad\approx \sum_{A,B_1,B_2,C_1,C_2,C_3} I(A = a_t)P(C_1, C_2|I_1, I_2)P(C_3|I_3)P(B_1|C_1, C_2)P(B_2|C_3)P(A|B_1, B_2)\\
&\quad\approx \sum_{A,B_1,B_2} I(A = a_t)P(A|B_1, B_2)\left(\sum_i P\left(B_1|C_1^i, C_2^i\right)\frac{P\left(I_1, I_2|C_1^i, C_2^i\right)}{\sum_i P\left(I_1, I_2|C_1^i, C_2^i\right)}\right)\left(\sum_j P\left(B_2|C_3^j\right)\frac{P\left(I_3|C_3^j\right)}{\sum_j P\left(I_3|C_3^j\right)}\right)\\
&\quad\approx \sum_{A,B_1,B_2} I(A = a_t)P(A|B_1, B_2)\sum_i\sum_j P\left(B_1, B_2|C_1^i, C_2^i, C_3^j\right)\frac{P\left(I_1, I_2|C_1^i, C_2^i\right)}{\sum_i P\left(I_1, I_2|C_1^i, C_2^i\right)}\frac{P\left(I_3|C_3^j\right)}{\sum_j P\left(I_3|C_3^j\right)}\\
&\quad\approx \sum_{A,i,j} I(A = a_t)\sum_k P\left(A|B_1^k, B_2^k\right)\frac{P\left(C_1^i, C_2^i, C_3^j|B_1^k, B_2^k\right)}{\sum_k P\left(C_1^i, C_2^i, C_3^j|B_1^k, B_2^k\right)}\frac{P\left(I_1, I_2|C_1^i, C_2^i\right)}{\sum_i P\left(I_1, I_2|C_1^i, C_2^i\right)}\frac{P\left(I_3|C_3^j\right)}{\sum_j P\left(I_3|C_3^j\right)}\\
&\quad\approx \sum_l I\left(A^l = a_t\right)\sum_k \frac{P\left(B_1^k, B_2^k|A^l\right)}{\sum_l P\left(B_1^k, B_2^k|A^l\right)}\sum_{i,j}\frac{P\left(C_1^i, C_2^i, C_3^j|B_1^k, B_2^k\right)}{\sum_k P\left(C_1^i, C_2^i, C_3^j|B_1^k, B_2^k\right)}\frac{P\left(I_1, I_2|C_1^i, C_2^i\right)}{\sum_i P\left(I_1, I_2|C_1^i, C_2^i\right)}\frac{P\left(I_3|C_3^j\right)}{\sum_j P\left(I_3|C_3^j\right)} \qquad (8)
\end{aligned}$$

For example, $P(C_3)$ is calculated by

$$\begin{aligned}
P(C_3) &= \sum_{B_2} P(B_2, C_3) = \sum_{B_2} P(B_2)P(C_3|B_2)\\
&= \frac{1}{K}\sum_{i=1}^{K} P\left(C_3|B_2^k\right). \quad B_2^k \sim P(B). \qquad (11)
\end{aligned}$$

IV. Theoretical Analysis of Conditional Independence Assumptions

To use sampling to optimize the posterior distributions required for inference, we have made a number of simplifying assumptions that enable the sampling to be local. It turns out that the simplifying assumptions are equivalent to conditional independence assumptions within the generative model (that could be regarded as a mean field approximation). We will take some care to illustrate the particular independence assumptions and the conditions under which they can be, effectively, ignored.

Here we consider the simple networks as in Fig. 2(a), of which the inference equation (8) is based on two sets of conditional independence assumptions.

Set 1: $B_1 \perp C_3|C_1, C_2$, $B_1 \perp B_2|C_3$, $B_1 \perp C_3^j|C_1^i, C_2^i$, and $B_1 \perp B_2|C_3^j$.

Set 2: $C_1, C_2 \perp I_3|I_1, I_2$ and $C_1, C_2 \perp C_3|I_3$.

The following theorems resolve these conditional independence assumptions, respectively. Specifically, we first prove by Theorem 1 that the assumptions in set 1 do not affect the accuracy of the inference algorithm, which means the inference results will converge to the accurate value with probability 1 as the sample size tends to infinity. Then we prove by Theorem 2 that the assumptions in set 2 hold approximately if the structure of the STM includes multilayers.

Theorem 1: Considering the Bayesian network shown in Fig. 3(a), we define that: $f_1(Y_1, Y_2) = \sum_{Z_1,Z_2} P(Y_1, Y_2|Z_1, Z_2)P(Z_1|T_1)P(Z_2|T_2)$, $f_2(Y_1, Y_2) = \sum_{i=1}^{M}\sum_{j=1}^{N} P(Y_1, Y_2|Z_1^i, Z_2^j)([P(T_1|Z_1^i)]/[\sum_{i=1}^{M} P(T_1|Z_1^i)])$ $([P(T_2|Z_2^j)]/[\sum_{j=1}^{N} P(T_2|Z_2^j)])$, $Z_1^i \sim P(Z_1)$, and $Z_2^j \sim P(Z_2)$, then for arbitrary small number ε, we have

$$\lim_{\substack{M\to\infty\\ N\to\infty}} P(|f_2(Y_1, Y_2) - f_1(Y_1, Y_2)| < \varepsilon) = 1. \qquad (12)$$

The proofs of Theorem 1 is in Appendix A. Theorem 1 shows that $f_2(Y_1, Y_2)$ is an estimator of $f_1(Y_1, Y_2)$ and converges to $f_1(Y_1, Y_2)$ with probability 1 when M and N tend to infinity. With Theorem 1, we can demonstrate that the conditional independent assumptions in set 1 will not affect the accuracy of the proposed algorithm. Specifically, the conditional independence assumptions used in (8) include the following four steps:

$$\begin{aligned}
g_1 &= \sum_A I(A = a_t)\sum_{B_1,B_2 P(A|B_1,B_2)}\sum_{C_1,C_2,C_3}\\
&\quad\times \{P(C_1, C_2|I_1, I_2)P(C_3|I_3)P(B_1, B_2|C_1, C_2, C_3)\} \qquad (13)\\
g_2 &= \sum_A I(A = a_t)\sum_{B_1,B_2} P(A|B_1, B_2)\sum_{C_1,C_2,C_3}
\end{aligned}$$

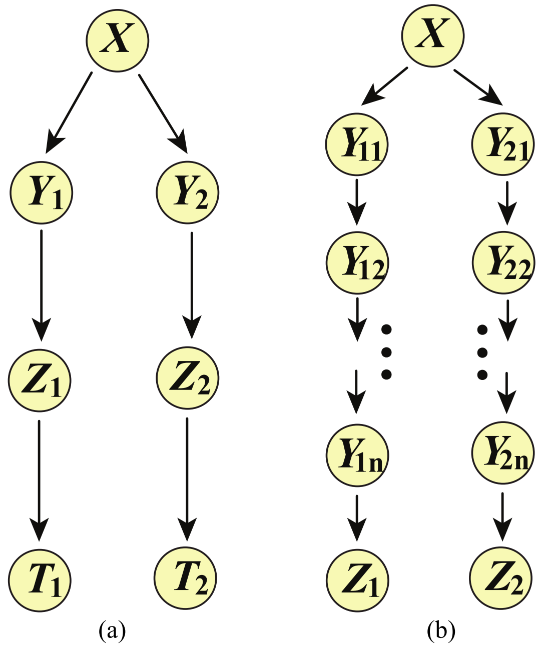


Fig. 3. Basic Bayesian models for illustrating conditional independence assumptions. (a) Simple Bayesian network for illustrating Theorem 1 and the conditional independence assumptions in set 1. (b) Multihierarchy Bayesian network for illustrating Theorem 2 and the conditional independence assumptions in set 2.

$$\times\{P(C_1, C_2|I_1, I_2)P(C_3|I_3)P(B_1|C_1, C_2)P(B_2|C_3)\} \tag{14}$$

$$\begin{aligned} g_3 = {} & \sum_A I(A = a_t) \sum_{B_1,B_2} \\ & \times \left\{ P(A|B_1, B_2) \left(\sum_i P\left(B_1|C_1^i, C_2^i\right) \frac{P\left(I_1, I_2|C_1^i, C_2^i\right)}{\sum_i P\left(I_1, I_2|C_1^i, C_2^i\right)} \right) \right. \\ & \left. \times \left(\sum_j P\left(B_2|C_3^j\right) \frac{P\left(I_3|C_3^j\right)}{\sum_j P\left(I_3|C_3^j\right)} \right) \right\} \\ & \times C_1^i, C_2^i \sim P(C_1, C_2) C_3^j \sim P(C_3) \end{aligned} \tag{15}$$

$$\begin{aligned} g_4 = {} & \sum_A I(A = a_t) \sum_{B_1,B_2} P(A|B_1, B_2) \sum_i \sum_j \\ & \times \left\{ P\left(B_1, B_2|C_1^i, C_2^i, C_3^j\right) \frac{P\left(I_1, I_2|C_1^i, C_2^i\right)}{\sum_i P\left(I_1, I_2|C_1^i, C_2^i\right)} \frac{P\left(I_3|C_3^j\right)}{\sum_j P\left(I_3|C_3^j\right)} \right\} \\ & \times C_1^i, C_2^i \sim P(C_1, C_2) C_3^j \sim P(C_3). \end{aligned} \tag{16}$$

The transformation from (13) to (14) includes the conditional independence assumptions $B_1 \perp C_3 | C_1, C_2$ and $B_1 \perp B_2 | C_3$. The transformation from (14) to (15) is based on importance sampling. Equation (16) includes the assumptions $B_1 \perp C_3^j | C_1^i, C_2^i$, and $B_1 \perp B_2 | C_3^j$. With Theorem 1, one can prove that for arbitrary small number ε, $\lim_{\substack{M\to\infty \\ N\to\infty}} P(|g_4 - g_1| < \varepsilon) = 1$ with M and N representing the sample sizes of C_1^i, C_2^i, and C_3^j, respectively.

The above results illustrate that the conditional independence assumptions in set 1 do not affect the accuracy of our algorithm. Thus we are able to regard (16) as a generalized importance sampling of (13). We show in the next section that this sampling-based inference process can be easily implemented by a network of neurons. However, the mathematical principles behind it are complex. The result is universal in our algorithm for different models as long as it includes structure as that in Fig. 3(a).

Theorem 2: Considering the Bayesian network shown in Fig. 3(b), the prior distribution $P(X)$ and conditional distribution $P(Z_t|Y_{t,n})$ are created by generated some numbers randomly from a uniform distribution on [0, 1] and then normalizing them ($t = 1, 2$). Similarly, the conditional distribution $P(Y_{t,1}|X)$ and $P(Y_{t,i+1}|Y_{t,i})$ are generated randomly and the probability of each state is nonzero ($i = 1, 2, \ldots, n-1$ and $t = 1, 2$), then we conclude that $Z_1 \perp Z_2$ when n tends to infinity.

The proof of Theorem 2 is in Appendix B. Theorem 2 shows that the dependence between Z_1 and Z_2 decrease as the hierarchy increases and will converge to zero if the hierarchy tends to infinity. We use this theorem to explain that the conditional independence assumptions in set 2 are reasonable. With Theorem 2, we can prove that the variables C_1–C_3 are approximately independent, which means $P(C_1, C_2, C_3) = P(C_1, C_2)P(C_3)$. Then, we can get

$$\begin{aligned} & P(C_1, C_2|I_1, I_2, I_3) \\ & \quad = \frac{\sum_{C_3} P(C_1, C_2, C_3, I_1, I_2, I_3)}{\sum_{C_1,C_2,C_3} P(C_1, C_2, C_3, I_1, I_2, I_3)} \\ & \quad = \frac{\sum_{C_3} P(C_1, C_2)P(C_3)P(I_1, I_2|C_1, C_2)P(I_3|C_3)}{\sum_{C_1,C_2} P(C_1, C_2)P(I_1, I_2|C_1, C_2) \sum_{C_3} P(C_3)P(I_3|C_3)} \\ & \quad = \frac{P(I_1, I_2, C_1, C_2)P(I_3)}{P(I_1, I_2)P(I_3)} \\ & \quad = P(C_1, C_2|I_1, I_2) \end{aligned} \tag{17}$$

and

$$\begin{aligned} P(C_3|C_1, C_2, I_3) & = \frac{P(C_1, C_2, C_3, I_3)}{\sum_{C_3} P(C_1, C_2, C_3, I_3)} \\ & = \frac{P(C_1, C_2)P(C_3)P(I_3|C_3)}{\sum_{C_3} P(C_1, C_2)P(C_3)P(I_3|C_3)} = P(C_3|I_3) \end{aligned} \tag{18}$$

which means $C_1, C_2 \perp I_3 | I_1, I_2$, C_1, and $C_2 \perp C_3 | I_3$. From the perspective of Bayesian networks, C_1–C_3 are not independent. However, this independence can happen in neuronal system as neuronal networks are hierarchical.

In conclusion, the hierarchical structure of the brain can ensure that some conditional independence assumptions are satisfied approximately, thus ensuring the accuracy of the inference algorithm. Now we have proved that our proposed STM can approximate Bayesian inference theoretically. The simulation experiments in the later section confirm this point.

V. Neural Network Implementation

In this section, we introduce the detailed neural network architecture of STM that can implement sampling-based inference algorithm. Shi and Griffiths [38] used radial basis function (RBF) networks to implement importance sampling and illustrated that the basic operations of the RBF model have neural correlates. However, they did not show how to calculate

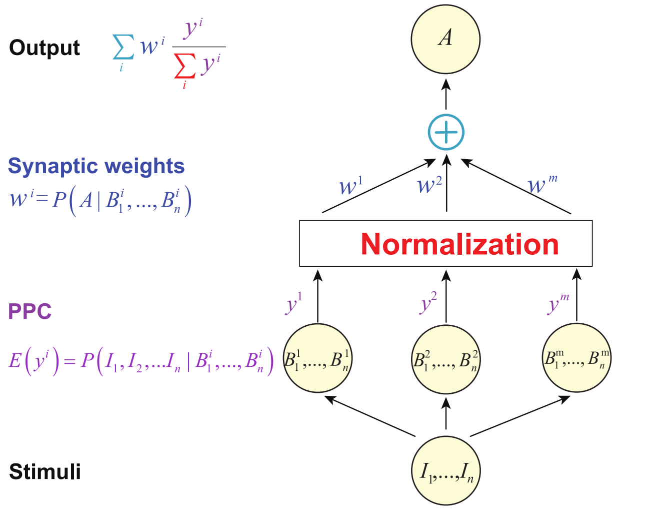

Output
$\sum_i w^i \frac{y^i}{\sum_i y^i}$
A
Synaptic weights
$w^i = P\left(A \mid B_1^i, \ldots, B_n^i\right)$
w^1
w^2
w^m
Normalization
PPC
$E\left(y^i\right) = P\left(I_1, I_2, \ldots I_n \mid B_1^i, \ldots, B_n^i\right)$
y^1
y^2
y^m
$B_1^1, \ldots, B_n^1$
$B_1^2, \ldots, B_n^2$
$B_1^m, \ldots, B_n^m$
Stimuli
$I_1, \ldots, I_n$

Fig. 4. Neural network architecture of the STM for the basic network as in Fig. 2(c). Computations done by this network are based on PPC (purple) and three types of biologically plausible operations: normalization (red), multiplication (blue), and linear combination (light blue).

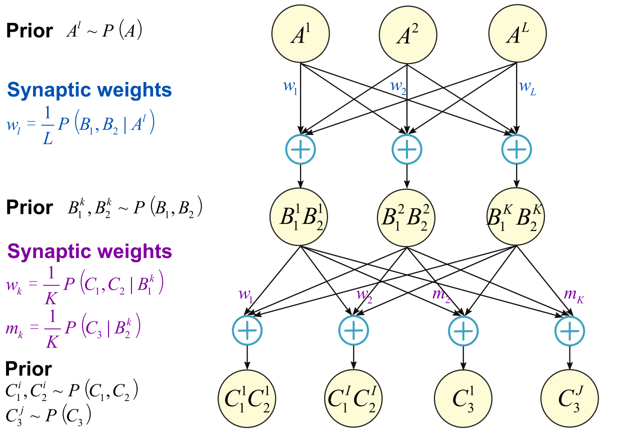

Prior $A^l \sim P(A)$
A^1
A^2
A^L
Synaptic weights
$w_l = \frac{1}{L} P\left(B_1, B_2 \mid A^l\right)$
w_1
w_2
w_L
Prior $B_1^k, B_2^k \sim P\left(B_1, B_2\right)$
$B_1^1 B_2^1$
$B_1^2 B_2^2$
$B_1^K B_2^K$
Synaptic weights
$w_k = \frac{1}{K} P\left(C_1, C_2 \mid B_1^k\right)$
$m_k = \frac{1}{K} P\left(C_3 \mid B_2^k\right)$
w_1
w_2
m_2
m_K
Prior
$C_1^i, C_2^i \sim P\left(C_1, C_2\right)$
$C_3^j \sim P\left(C_3\right)$
$C_1^1 C_2^1$
$C_1^I C_2^I$
C_3^1
C_3^J

Fig. 5. Top-down process of calculating prior probabilities. This neural network architecture of the STM is used to calculate prior probabilities of the Bayesian model in Fig. 2(a).

prior probabilities. In this paper, we will calculate prior probabilities and implement inference in the similar network based on PPC and several biologically plausible operations. We first show how to implement inference in basic network motif with simple STM. Then we use serial and parallel combination of these basic networks to build a large-scale STM to calculate prior probabilities from top to down and perform inference for arbitrary tree-structured Bayesian model from bottom to up.

Before we give detailed circuits of STM, we give a brief introduction of PPC. PPC takes advantages of the variability in neuronal responses and considers that a population of neurons can encode the probability distributions, instead of the values of variables. Specifically, for N independent Poisson spiking neurons, the distribution of the responses $r = \{r_1, r_2, \ldots, r_N\}$ to the input stimulus S is $P(r|S) = \prod_i [(e^{-f_i(s)} f_i(s)^{r_i})/(r_i!)]$, where $f_i(s)$ represents the tuning curve of the neuron i and is a function of the input stimulus S, which represents the average firing rate of stimulus S over an infinite number of trials. With this definition, the distribution of the input stimulus S is encoded by the neural activities $r = \{r_1, r_2, \ldots, r_N\}$.

Fig. 4 shows the neural network layout of the STM to implement inference for our basic network motif as in Fig. 2(c), which includes PPC and three types of plausible neural operations: 1) normalization; 2) multiplication; and 3) linear combination that can be realized by computation in neural circuits [50]. To be specific, there are m Poisson spiking neurons, each of which has a specific attribute, like tuning curve, and can represent a specific state of variables $B_1, B_2, \ldots, B_n$. The distributions of these Poisson spiking neurons follows the prior distribution $P(B_1, B_2, \ldots, B_n)$, and the tuning curve of the neuron i is supposed to be proportional to the conditional distribution $P(I_1, I_2, \ldots, I_n | B_1^i, B_2^i, \ldots, B_n^i)$, where $I_1, I_2, \ldots, I_n$ are input stimuli. Note that the prior and conditional distributions are known. The output of Poisson spiking neurons are normalized by shunting inhibition and/or synaptic depression [38], [51], [52] (refer to [53, Fig. 1] for detailed neural circuit). If we use y_i to express the individual output firing rate of Poisson spiking neuron i and Y to express the total firing rate, i.e., $Y = \sum_i y_i$, then

$$E(y_i/Y = n) = \frac{P\big(I_1, I_2, \ldots, I_2 | B_1^i, B_2^i, \ldots, B_n^i\big)}{\sum_i P\big(I_1, I_2, \ldots, I_2 | B_1^i, B_2^i, \ldots, B_n^i\big)} \tag{19}$$

which is proved in [38]. This result shows the expectation of the individual firing rate relative to total firing rate equals to normalized conditional probability. The normalized results are linearly combined with their synaptic weights $w_i = P(A|B_1^i, B_2^i, \ldots, B_n^i)$ to get a summation output as

$$\begin{aligned} E\left(\sum_i w_i y_i / Y = n\right) &= \sum_i w_i E(y_i/Y = n) \\ &= \sum_i P\big(A|B_1^i, B_2^i, \ldots, B_n^i\big) \\ &\quad \frac{P\big(I_1, I_2, \ldots, I_2 | B_1^i, B_2^i, \ldots, B_n^i\big)}{\sum_i P\big(I_1, I_2, \ldots, I_2 | B_1^i, B_2^i, \ldots, B_n^i\big)} \end{aligned} \tag{20}$$

which equals to the inference result in (2).

Next, we illustrate a large neural network with a few more components of the STM for the Bayesian model in Fig. 2(a). The two processes are shown in Fig. 5 for the top-down process of calculating prior probabilities and Fig. 6 for the bottom-up process of performing inference. The whole neural network is the serial and parallel combinations of the basic network motifs.

We first discuss the top-down process as shown in Fig. 5. There are feature detection neurons $A_1, A_2, \ldots, A_L$ with their states proportional to the prior distribution $P(A)$. Supposing that the synaptic weight to the next layer is $P(B_1, B_2|A_l)/L$, the probability $P(B_1, B_2)$ could be calculated by $P(B_1, B_2) = (1/L) \sum_{l=1}^{L} P(B_1, B_2|A_l)$. The states of feature detection neurons in the next layer are then decided by the probability $P(B_1, B_2)$. The probabilities $P(C_1, C_2)$ and $P(C_3)$ could be calculated in a similar way. This top-down process calculate all the prior probabilities $P(B_1, B_2)$, $P(C_1, C_2)$, and $P(C_3)$ and ensure that the frequencies of these feature detection neurons are proportional to the prior probabilities.

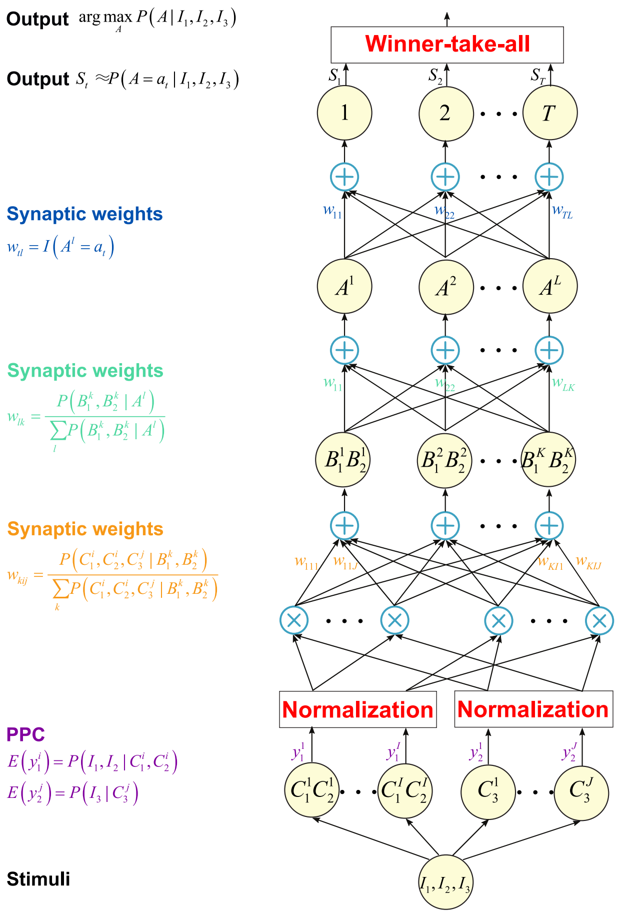


Fig. 6. Bottom-up process of performing inference. This neural network architecture of the STM is used to implement inference of the Bayesian model in Fig. 2(a).

On the contrary, the inference process is bottom-up as shown in Fig. 6. There are I Poisson spiking neurons that encode the variables C_1 and C_2 and J Poisson spiking neurons that encode the variable C_3. The distribution of these Poisson spiking neurons follows the prior distributions $P(C_1, C_2)$ and $P(C_3)$. Besides, the tuning curves are proportional to $P(I_1, I_2|C_1^i, C_2^i)$ and $P(I_3|C_3^j)$ respectively. The responses of Poisson spiking neurons in the bottom layer are normalized by shunting inhibition and/or synaptic depression [38], [51], [52]. Using the conclusion in [38], we can get that the expectation of mean firing rates of the Poisson spiking neurons C_1^i, C_2^i, and C_3^j are $P(I_1, I_2|C_1^i, C_2^i)/\sum_i P(I_1, I_2|C_1^i, C_2^i)$ and $P(I_3|C_3^j)/\sum_j P(I_3|C_3^j)$, respectively. These firing rates are multiplied together and fed into the next layer with the synaptic weight $P(C_1^i, C_2^i, C_3^j|B_1^k, B_2^k)/\sum_k P(C_1^i, C_2^i, C_3^j|B_1^k, B_2^k)$ and the outputs of the neurons are $\sum_{i,j}[(P(C_1^i, C_2^i, C_3^j|B_1^k, B_2^k))/(\sum_k P(C_1^i, C_2^i, C_3^j|B_1^k, B_2^k))]$, $[(P(I_1, I_2|C_1^i, C_2^i))/(\sum_i P(I_1, I_2|C_1^i, C_2^i))]$, and $[(P(I_3|C_3^j))/(\sum_j P(I_3|C_3^j))]$, where B_1^k and B_2^k are feature detection neurons with their states proportional to the prior probability $P(B_1, B_2)$. The process is similar in other layers and we can get the posterior probability $P(A|I_1, I_2, I_3)$ in the fourth layer, which equals to the result in (8). Based on this, MAP estimation $\arg\max_A P(A|I_1, I_2, I_3)$ is easy to be calculated since we only need to add a winner-take-all (WTA) circuit[2] after the fourth layer.

The STM has the feature that most of computations are done by simple neural network motifs. Therefore, it uses massive number of neurons to sample in parallel and calculates only once without iterations, for instance, the STM can use a thousand neurons to sample one time instead of a neuron sampling a thousand times. As a result, the inference is quite fast and efficient. The apparent cost is that the STM needs a large number of neurons. Luckily, there are about 80 billion neurons in human brain, which seems to be reasonable enough for parallel computing, similar to the computational principle of our proposed STM.

VI. Simulations

We test the accuracy of the STM for Bayesian inference on two cognitive problems: 1) the integration of multicue information and 2) object detection with compositional model. The first one is a benchmark problem used to test the accuracy of Bayesian inference method. The second one is a larger and more complex problem, and it is used to examine whether our method can scale up to large-scale Bayesian model.

A. *Integration of Multicue Information*

In our daily life, we often receive sensory information from vision, hearing and tough simultaneously. Experimental evidence shows that the human brain is able to integrate them in a Bayesian style [55]. At the neuronal level, Ma *et al.* [32] explained that linear combinations of different neuronal population activities with PPC correspond to the process of cue integration. Here we show that our proposed STM can solve multicue integration with a high accuracy.

The haptic-visual-auditory integration problem is considered in this paper, which could be modeled by the Bayesian network shown in Fig. 7(a). Here S, S_H, S_V, and S_A denote the location of the stimulus, haptic, visual, and auditory cues, respectively. Supposing that $P(S)$ is a uniform distribution, $P(S_H|S)$, $P(S_V|S)$ and $P(S_A|S)$ are three different Gaussian distributions with the same mean value S and different variances $\sigma_{S_H}^2$, $\sigma_{S_V}^2$, and $\sigma_{S_A}^2$, then we can infer the posterior probability of S given S_H, S_V, and S_A with importance sampling

$$\begin{aligned} P(S = s|S_H, S_V, S_A) &= \sum_S I(S = s)P(S|S_H, S_V, S_A) \\ &= \sum_i I(S_i = s)\frac{P(S_H, S_V, S_A|S_i)}{\sum_i P(S_H, S_V, S_A|S_i)} S_i \sim P(S). \end{aligned} \tag{21}$$

In our simulation, there are 5000 Poisson spiking neurons, the states of which follow the distribution $P(S)$. The tuning curve of the neuron i is supposed to be proportional to the distribution $P(S_H, S_V, S_A|S_i)$. The output of Poisson spiking neurons are normalized by shunting inhibition and/or synaptic depression. The normalized results are fed into the output

[2] WTA circuit is an ubiquitous motif of cortical microcircuits in the brain, which consists of ensemble of excitatory cells with lateral inhibition [54]. With the competition between excitatory cells induced by the inhibition, only the excitatory neuron with the largest membrane can fire.

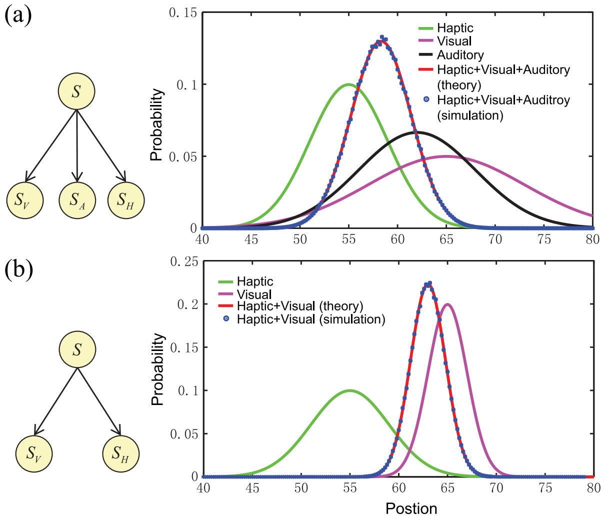


Fig. 7. Simulation of multicue integration. (a) Left: a Bayesian model for haptic (green line), visual (purple line), and auditory (black line) integration, and right: comparison of the inference results with STM (blue dots) and theoretical value (red line). $\sigma^2_{S_H} = 64$, $\sigma^2_{S_V} = 16$, and $\sigma^2_{S_A} = 36$. Each point is averaged over ten trials. (b) Similar to (a) but for visual-auditory integration. $\sigma^2_{S_H} = \sigma^2_{S_V} = 16$.

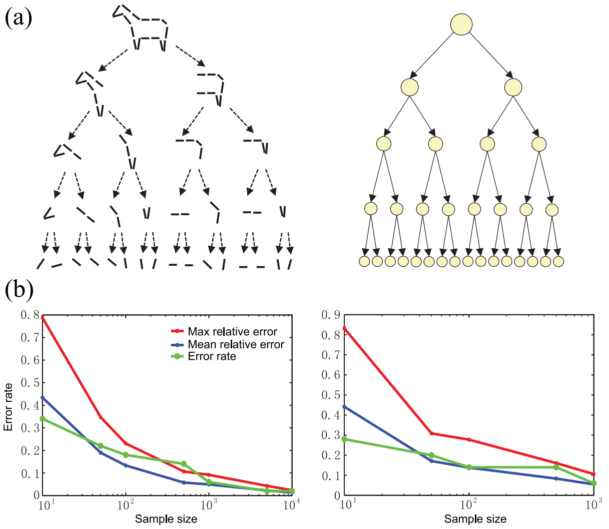


Fig. 8. Object detection with a large scale Bayesian model. (a) Left: example of horse can be decomposed into smaller parts layer by layer with a compositional model. Right: represented Bayesian model. (b) Simulation of three-layer (left) and four-layer (right) compositional models. Max relative error, mean relative error, and error rate decay to zero when sample size is large enough.

neuron with the synaptic weights $I(S_i = s)$. Fig. 7(a) illustrates the experimental results, where the inference results obtained by STM match the theoretical values very well. Similar to the previous study [38], the case of two-cue integration is illustrated in Fig. 7(b) for the completeness.

B. Object Detection With Compositional Model

Now we test our sampling-based inference algorithm for large-scale Bayesian model with a compositional model for object detection. Compositional model is a generative model which represents objects similar to human brain [56]–[58]. It assumes that an object can be decomposed into small parts and these parts can be decomposed into smaller and smaller parts until we get the smallest parts, such as the horizontal and vertical lines. The process of object detection is on the contrary, which starts from detecting the smallest parts of the picture and then composes these parts to detect the bigger one until the whole object is detected. A typical example of the compositional model is shown in Fig. 8(a); a horse can be divided into two small parts and each part can be divided into smaller parts, until we get the basic lines. If we want to use the model to detect the location of a horse in a picture, we first detect all the basic lines, then compose these lines to infer the location of a bigger part and for the same to the horse at last.

The compositional model can be modeled by Bayesian networks. Specifically, every node in the Bayesian network represents a special part of the object. A parent node v represents a part of the object. It has r children nodes $Ch(v) = (v_1, v_2, \ldots, v_r)$, which represent r compositional parts of the bigger part. Besides, each node has random variables attached to it, which is specified by x, reflecting the location of the part. Similarly, the variables attached to the children nodes are $x_{\mathrm{Ch}(v)} = (x_{v_1}, x_{v_1}, \ldots, x_{v_r})$; here $x_{v_1}, x_{v_1}, \ldots, x_{v_r}$ are the location of the r compositional parts. Supposing that the total hierarchy of the model is H and the total nodes are V, it is easy to see that $V = V_1 \cup V_2 \cup \cdots \cup V_H$, where $V_1, V_2, \ldots, V_H$ are nodes attached to each level. The prior probability of node in H is defined by $P(x_H)$. Here we suppose that there is only one node in the highest level H, which represents the object, and the distribution of variable x_H is uniform. The conditional probability distribution of the children nodes under the condition of the parent node is $P(x_{\mathrm{Ch}(v)}|x_v)$.

With the definitions above, the probability distribution of the model can be computed by

$$P(x) = \left(\prod_{v \in V/V_1} P\big(x_{\mathrm{Ch}(v)}|x_v\big)\right) P(x_H). \tag{22}$$

Suppose that the nodes in the lowest level of the model are connected to the image directly, and then the conditional probability distribution of the image given the state of these nodes is

$$P(I|x) = \prod_{v \in V_1} P(I(x_v)|x_v) \tag{23}$$

where I is the input image and $P(I(x_v)|x_v)$ is probability of the image conditioned on the nodes in the lowest level. As the problem is to detect the location of an object, the inference problem is $x_H = \arg\max_{x_H} P(x_H|I)$, that is, inferring the state of the root node given the input variables.

The represented Bayesian model for the horse is in Fig. 8(a). The root node represents the location of the horse and the leaf nodes represent the locations of the basic lines in the picture. Here we calculate posterior probability $P(x_H|I)$ with STM and express the result as $P^{STM}(x_H = i|I)$ $(i = 1, 2, \ldots, N)$, where N represents the number of all possible states of variable x_H. Meanwhile, the truth of $P(x_H|I)$ is expressed as $P^{\mathrm{truth}}(x_H = i|I)$ $(i = 1, 2, \ldots, N)$, which is calculated with the elimination method. The relative error is defined as $[(|P^{\text{STM}}(x_H = i|I) - P^{\text{truth}}(x_H = i|I)|)/(P^{\text{truth}}(x_H = i|I))]$ ($i = 1, 2, \ldots, N$). Fig. 8(b) shows the simulation results for three- and four-layer compositional models, where max relative error is the maximum value of the relative error for all state ($i = 1, 2, \ldots, N$) of the root node, mean relative error expresses the mean value of the relative error for all states of the root node. The error rate is the accuracy rate when we calculate the maximum *a posterior* with our method compared to the true value. These three indexes show inference accuracy of our method based on STM. All these errors decrease as sample size increase and will be close to zero when sample size is large enough. Therefore, these experimental results show that our method can get accurate inference for large-scale Bayesian models.

VII. Conclusion

It is of great importance to understand how the brain performs Bayesian inference with a network of neurons. In this paper, we proposed a sampling-based inference model, termed STM, which is a distributed neural network that can implement fast and accurate inference of the arbitrary Bayesian model.

Our method is composed of a set of simple and basic neural network motifs, and uses a massive number of neurons to sample in parallel and perform computation locally in space. For example, our method can use a set of 1000 neurons to sample one time instead of a single neuron to sample 1000 times. As a result, the inference is quite fast. The apparent cost is that our method needs large numbers of neurons for sampling. Considering the fact that there are billions of neurons in the brain, and we do perform reasoning quite fast, our method suggests a plausible way for neural implementation of our cognitive behaviors.

With the great advancements of recent hardwares, including neuromorphic chips, it is expected that our method can be implemented with both artificial neural networks and spiking neural networks, and that is a direction we are pursuing. The hardware also provides the basis for large-scale distributed Bayesian inference, which is the main feature of our algorithm.

Although most of current neuroscience experiments are conducted for relatively simple cognition behaviors, some more complex tasks have been proposed, for example, a hierarchical decision-making task [59]. In future work, we will explore these complex tasks with a large-scale of Bayesian network based on our model.

Another important aspect we did not consider here is learning [60]. Here all the results are based on the condition that we have known prior probabilities and conditional probabilities. In fact, our brain does have the ability to learn the probabilities and update them in time [24]. Some recent works have provided reference experiences for unsupervised learning [61], supervised learning [62] and reward-based learning [63] of the brain, which may be used to solve the learning problem in this paper. Besides, how to combine learning with inference is an active research direction [64], [65]. Future work is needed to unify our method and some learning mechanisms, like spike-timing-dependent plasticity [66], [67], into one framework.

Appendix A
Proof of Theorem 1

Theorem 1: Considering the Bayesian network shown in Fig. 3(a), we define that: $f_1(Y_1, Y_2) = \sum_{Z_1, Z_2} P(Y_1, Y_2|Z_1, Z_2)P(Z_1|T_1)P(Z_2|T_2)$, $f_2(Y_1, Y_2) = \sum_{i=1}^{M}\sum_{j=1}^{N} P(Y_1, Y_2|Z_1^i, Z_2^j)[(P(T_1|Z_1^i))/(\sum_{i=1}^{M} P(T_1|Z_1^i))]$, $[(P(T_2|Z_2^j))/(\sum_{j=1}^{N} P(T_2|Z_2^j))]$, $Z_1^i \sim P(Z_1)$, and $Z_2^j \sim P(Z_2)$, then for an arbitrary small number ε, we have

$$\lim_{\substack{M\to\infty\\ N\to\infty}} P\left(|f_2(Y_1, Y_2) - f_1(Y_1, Y_2)| < \varepsilon\right) = 1. \tag{24}$$

Proof: We rewrite $f_2(Y_1, Y_2)$ as

$$\begin{aligned}
&f_2(Y_1, Y_2)\\
&\quad= \sum_{i=1}^{M}\sum_{j=1}^{N} P\Big(Y_1, Y_2|Z_1^i, Z_2^j\Big)\frac{P\big(T_1|Z_1^i\big)}{\sum_{i=1}^{M} P\big(T_1|Z_1^i\big)}\frac{P\Big(T_2|Z_2^j\Big)}{\sum_{j=1}^{N} P\Big(T_2|Z_2^j\Big)}\\
&\qquad\qquad\qquad\qquad\qquad\qquad Z_1^i \sim P(Z_1)\\
&\qquad\qquad\qquad\qquad\qquad\qquad Z_2^j \sim P(Z_2)\\
&\quad= \frac{\frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N} P\Big(Y_1, Y_2|Z_1^i, Z_2^j\Big)P\big(T_1|Z_1^i\big)P\Big(T_2|Z_2^j\Big)}{\frac{1}{MN}\sum_{k=1}^{M}\sum_{l=1}^{M} P\big(T_1|Z_1^k\big)P\big(T_2|Z_2^l\big)}\\
&\qquad\qquad\qquad\qquad Z_1^i \sim P(Z_1)Z_2^j \sim P(Z_2)\\
&\qquad\qquad\qquad\qquad Z_1^k \sim P(Z_1)Z_2^l \sim P(Z_2).
\end{aligned} \tag{25}$$

The expectation and variance take the form

$$\begin{aligned}
&E\Big(P\Big(Y_1, Y_2|Z_1^i, Z_2^j\Big)P\big(T_1|Z_1^i\big)P\Big(T_2|Z_2^j\Big)\Big)\\
&\quad= \sum_{Z_1^i}\sum_{Z_2^j}\Big\{P\Big(Y_1, Y_2|Z_1^i, Z_2^j\Big)P\big(T_1|Z_1^i\big)P\Big(T_2|Z_2^j\Big)\\
&\qquad\qquad \times P\big(Z_1^i\big)P\Big(Z_2^j\Big)\Big\}\\
&\quad= \sum_{Z_1}\sum_{Z_2} P(Y_1, Y_2|Z_1, Z_2)P(T_1, Z_1)P(T_2, Z_2)\\
&\quad= f_1(Y_1, Y_2)P(T_1)P(T_2)
\end{aligned} \tag{26}$$

$$\begin{aligned}
&E\Big(P\big(T_1|Z_1^k\big)P\big(T_2|Z_2^l\big)\Big)\\
&\quad= \sum_{Z_1^k}\sum_{Z_2^l} P\big(T_1|Z_1^k\big)P\big(T_2|Z_2^l\big)P\big(Z_1^k\big)P\big(Z_2^l\big)\\
&\quad= P(T_1)P(T_2)
\end{aligned} \tag{27}$$

$$\begin{aligned}
&\mathrm{Var}\Big(P\Big(Y_1, Y_2|Z_1^i, Z_2^j\Big)P\big(T_1|Z_1^i\big)P\Big(T_2|Z_2^j\Big)\Big)\\
&\quad= E\bigg(\Big(P\Big(Y_1, Y_2|Z_1^i, Z_2^j\Big)P\big(T_1|Z_1^i\big)P\Big(T_2|Z_2^j\Big)\Big)^2\bigg)\\
&\qquad - E\Big(P\Big(Y_1, Y_2|Z_1^i, Z_2^j\Big)P\big(T_1|Z_1^i\big)P\Big(T_2|Z_2^j\Big)\Big)^2\\
&\quad= \sum_{Z_1}\sum_{Z_2}\Big\{P(Y_1, Y_2|Z_1, Z_2)^2 P(T_1|Z_1)^2 P(T_2|Z_2)^2\\
&\qquad\qquad \times P(Z_1)^2 P(Z_2)^2\Big\}\\
&\qquad - f_1(Y_1, Y_2)^2 P(T_1)^2 P(T_2)^2
\end{aligned} \tag{28}$$

$$\begin{aligned}
&\mathrm{Var}\Big(P\Big(T_1|Z_1^k\Big)P\Big(T_2|Z_2^l\Big)\Big)\\
&\quad= E\Big(\Big(P\Big(T_1|Z_1^k\Big)P\Big(T_2|Z_2^l\Big)\Big)^2\Big) - E\Big(P\Big(T_1|Z_1^k\Big)P\Big(T_2|Z_2^l\Big)\Big)^2\\
&\quad= \sum_{Z_1}\sum_{Z_2} P(T_1|Z_1)^2 P(T_2|Z_2)^2 P(Z_1)P(Z_2) - P(T_1)^2 P(T_2)^2.
\end{aligned} \tag{29}$$

Since $f_1(Y_1, Y_2)P(T_1)P(T_2)/P(T_1)P(T_2) = f_1(Y_1, Y_2)$, it is easy to use [68, Lemma 1] to show that for an arbitrary small number ε, $\lim_{\substack{M\to\infty\\ N\to\infty}} P\ (|f_2(Y_1, Y_2) - f_1(Y_1, Y_2)| < \varepsilon) = 1$. ■

Appendix B

Lemma 1: Suppose that $A_1, A_2, \ldots, A_n$ is randomly generated matrices, and $\mathrm{row}(A_i) = \mathrm{col}(A_{i+1})$ holds for $i = 1, 2, \ldots, n$. Each element of the matrices $A_1, A_2, \ldots, A_n$ is in $[\varepsilon, 1 - \varepsilon]$, where ε is a small number. Besides, the sum of each row of the matrices $A_1, A_2, \ldots, A_n$ is 1. If one defines that $C_k = (\prod_{i=1}^{k} A_i^T)^T$, one can conclude that all elements in a special col of C_k will tend to the same value when k tends to infinity.

Proof of Theorem 2: It is easy to prove $C_i = A_i C_{i-1}$ if $i \geq 2$ and $C_i = A_i$ if $i = 1$. Besides, $\mathrm{col}(C_i) = \mathrm{col}(A_1)$ and $\mathrm{row}(C_i) = \mathrm{row}(A_i)$. Suppose that

$$A_i = \begin{bmatrix} a_{i,1,1} & a_{i,1,2} & \cdots & a_{i,1,n(i)}\\ a_{i,2,1} & a_{i,2,2} & \cdots & a_{i,2,n(i)}\\ \cdot & \cdot & \cdot & \cdot\\ \cdot & \cdot & \cdot & \cdot\\ a_{i,m(i),1} & a_{i,m(i),2} & \cdots & a_{i,m(i),n(i)} \end{bmatrix}$$

$$C_i = \begin{bmatrix} c_{i,1,1} & c_{i,1,2} & \cdots & c_{i,1,n(1)}\\ c_{i,2,1} & c_{i,2,2} & \cdots & c_{i,2,n(1)}\\ \cdot & \cdot & \cdot & \cdot\\ \cdot & \cdot & \cdot & \cdot\\ c_{i,m(i),1} & c_{i,m(i),2} & \cdots & c_{i,m(i),n(1)} \end{bmatrix}$$

where $m(i)$ and $n(i)$ represents the row and col of the matrix A_i. If one uses $\widehat{c_{i,j}}$ to express the vector of all the elements in col j of matrix C_i, then $\max(\widehat{c_{i,j}})$ represents the maximum element in col j of matrix C_i and $\min(\widehat{c_{i,j}})$ represents the minimum element in col j of matrix C_i. Now for arbitrary $c_{i+1,s,t}$, where $s \in (1, 2, \ldots, m(i+1))$, $t \in (1, 2, \ldots, n(1))$, we can get

$$\begin{aligned} c_{i+1,s,t} &= a_{i+1,s,1}c_{i,1,t} + a_{i+1,s,2}c_{i,2,t} + \cdots\\ &\quad + a_{i+1,s,n(i+1)}c_{i,m(i+1),t}. \end{aligned} \tag{30}$$

As $\sum_{j=1}^{n(i+1)} a_{i+1,s,j} = 1$, (30) is the weighted average of col t of matrix C_i. By using the condition that the arbitrary element of $A_1, A_2, \ldots, A_n$ is in $[\varepsilon, 1 - \varepsilon]$, one obtains

$$\begin{aligned} (1-\varepsilon)\min\big(\widehat{c_{i,t}}\big) + \varepsilon\max\big(\widehat{c_{i,t}}\big) &\leq c_{i+1,s,t}\\ \leq \varepsilon\min\big(\widehat{c_{i,t}}\big) + (1-\varepsilon)\max\big(\widehat{c_{i,t}}\big) \end{aligned} \tag{31}$$

which is equivalent to

$$\begin{aligned} 0 &\leq \max\big(\widehat{c_{i+1,t}}\big) - \min\big(\widehat{c_{i+1,t}}\big)\\ &\leq (1-2\varepsilon)\big(\max\big(\widehat{c_{i,t}}\big) - \min\big(\widehat{c_{i,t}}\big)\big). \end{aligned} \tag{32}$$

Equation (32) can be rewritten as

$$\begin{aligned} 0 &\leq \max\big(\widehat{c_{i+1,t}}\big) - \min\big(\widehat{c_{i+1,t}}\big)\\ &\leq (1-2\varepsilon)^i\big(\max\big(\widehat{c_{1,t}}\big) - \min\big(\widehat{c_{1,t}}\big)\big). \end{aligned} \tag{33}$$

If we compute the limitation for both sides of (33) when i tends to infinity, we obtain

$$\lim_{i\to\infty}\big(\max\big(\widehat{c_{i+1,t}}\big) - \min\big(\widehat{c_{i+1,t}}\big)\big) = 0 \tag{34}$$

which means that all elements in a special col of C_i will tend to the same value.

Theorem 2: Considering the Bayesian network shown in Fig. 3(b), the prior distribution $P(X)$ and the conditional distribution $P(Z_t|Y_{t,n})$ are created by generating some numbers randomly from a uniform distribution on [0, 1] and then normalizing them ($t = 1, 2$). Similarly, the conditional distributions $P(Y_{t,1}|X)$ and $P(Y_{t,i+1}|Y_{t,i})$ are generated randomly and the probability of each state is nonzero ($i = 1, 2, \ldots, n-1$ and $t = 1, 2$), then we conclude that $Z_1 \perp Z_2$ when n tends to infinity.

Proof: Suppose that $U_{t,1}$ ($t = 1$ or 2) is a matrix with its element in row i and col j expressed as $u_{t,1,i,j}$, and $u_{t,1,i,j} = P(Y_{t,1} = Y_{t,1}(j)|X = X(i))$, where $Y_{t,1}(j)$ stands for the jth element of variable $Y_{t,1}$ and $X(i)$ stands for the ith element of variable X. Similarly, $U_{t,s}$ ($t = 1$ or 2 and $s = 1, 2, n$) is a matrix with its element in row i and col j expressed as $u_{t,s,i,j}$, and $u_{t,s,i,j} = P(Y_{t,s} = Y_{t,s}(j)|Y_{t,s-1} = Y_{t,s-1}(i))$. Moreover, $U_{t,n+1}$ ($t = 1$ or 2) is a matrix with its element in row i and col j expressed as $u_{t,n+1,i,j}$, and $u_{t,n+1,i,j} = P(Z_{t,1} = Z_{t,1}(j)|Y_{t,n} = Y_{t,n}(i))$, then we have

$$\begin{aligned} P(Z_1) &= \sum_X\sum_{Y_{1,1}}\sum_{Y_{1,2}}\cdots\sum_{Y_{1,n}} P(X)P\big(Y_{1,1}|X\big)P\big(Y_{1,2}|Y_{1,1}\big)\cdots\\ &\qquad P\big(Y_{1,n}|Y_{1,n-1}\big)P\big(Z_1|Y_{1,n}\big)\\ &= \sum_X P(X)\sum_{Y_{1,1}} P\big(Y_{1,1}|X\big)\sum_{Y_{1,2}} P\big(Y_{1,2}|Y_{1,1}\big)\cdots\\ &\qquad \sum_{Y_{1,n}} P\big(Y_{1,n}|Y_{1,n-1}\big)P\big(Z_1|Y_{1,n}\big)\\ &= \sum_X P(X)f(X, Z_1). \end{aligned} \tag{35}$$

Similarly

$$\begin{aligned} P(Z_2) &= \sum_X\sum_{Y_{2,1}}\sum_{Y_{2,2}}\cdots\sum_{Y_{2,n}} P(X)P\big(Y_{2,1}|X\big)P\big(Y_{2,2}|Y_{2,1}\big)\cdots\\ &\qquad P\big(Y_{2,n}|Y_{2,n-1}\big)P\big(Z_2|Y_{2,n}\big)\\ &= \sum_X P(X)\sum_{Y_{2,1}} P\big(Y_{2,1}|X\big)\sum_{Y_{2,2}} P\big(Y_{2,2}|Y_{2,1}\big)\cdots\\ &\qquad \sum_{Y_{2,n}} P\big(Y_{2,n}|Y_{2,n-1}\big)P\big(Z_2|Y_{2,n}\big)\\ &= \sum_X P(X)g(X, Z_2) \end{aligned} \tag{36}$$

$$\begin{aligned} P(Z_1, Z_2) &= \sum_X\sum_{Y_{1,1}}\sum_{Y_{1,2}}\cdots\sum_{Y_{1,n}}\sum_{Y_{2,1}}\cdots\sum_{Y_{2,n}}\\ &\qquad \times\big\{P(X)P\big(Y_{1,1}|X\big)\cdots\\ &\qquad P\big(Y_{1,n}|Y_{1,n-1}\big)P\big(Z_1|Y_{1,n}\big)P(X)P\big(Y_{2,1}|X\big)\\ &\qquad \times P\big(Y_{2,2}|Y_{2,1}\big)\cdots ft.P\big(Y_{2,n}|Y_{2,n-1}\big)P\big(Z_2|Y_{2,n}\big)\big\}\\ &= \sum_X P(X)\sum_{Y_{1,1}} P\big(Y_{1,1}|X\big)\sum_{Y_{1,2}} P\big(Y_{1,2}|Y_{1,1}\big)\cdots\\ &\qquad \sum_{Y_{1,n}} P\big(Y_{1,n}|Y_{1,n-1}\big)P\big(Z_1|Y_{1,n}\big)\sum_{Y_{2,1}} P\big(Y_{2,1}|X\big) \end{aligned}$$

$$\begin{aligned}&\times \sum_{Y_{2,2}} P\big(Y_{2,2}|Y_{2,1}\big) \cdots \sum_{Y_{2,n}} P\big(Y_{2,n}|Y_{2,n-1}\big) P\big(Z_2|Y_{2,n}\big)\\ &= \sum_X P(X) f(X, Z_1) g(X, Z_2) \qquad (37)\end{aligned}$$

where $f(X = i, Z_1 = j)$ denotes the element in the ith row and jth col of the matrix $\prod_{i=1}^{n+1} U_{1,i}$, and $g(X = i, Z_2 = j)$ denotes the element in the ith row and jth col of the matrix $\prod_{i=1}^{n+1} U_{2,i}$. When n goes to infinity, we can prove that all the elements in each col of $\prod_{i=1}^{n+1} U_{1,i}$ ($\prod_{i=1}^{n+1} U_{2,i}$) tend to the same value by using Lemma 2. It means that $f(X, Z_1)$ and $g(X, Z_2)$ are independent of X, respectively. In other words, $f(X, Z_1) \approx f_1(Z_1)$ and $g(X, Z_2) \approx g_1(Z_2)$. Above all, when n goes to infinity, one obtains

$$\begin{aligned}P(Z_1, Z_2) &= \sum_X P(X) f(X, Z_1) g(X, Z_2)\\ &= \sum_X P(X) f_1(Z_1) g_1(Z_2)\\ &= f_1(Z_1) g_1(Z_2)\\ &= \left(\sum_X P(X) f_1(Z_1)\right)\left(\sum_X P(X) g_1(Z_2)\right)\\ &= \left(\sum_X P(X) f(X, Z_1)\right)\left(\sum_X P(X) g(X, Z_2)\right)\\ &= P(Z_1) P(Z_2) \qquad (38)\end{aligned}$$

which means $Z_1 \perp Z_2$ as n tends to infinity. ■

ACKNOWLEDGMENT

The authors would like to thank J. Dong, Y. Gao, and Y. Zhou for helpful discussions.

REFERENCES

[1] A. Pouget, J. M. Beck, W. J. Ma, and P. E. Latham, "Probabilistic brains: Knowns and unknowns," *Nat. Neurosci.*, vol. 16, no. 9, pp. 1170–1178, 2013.

[2] F. Meyniel, M. Sigman, and Z. F. Mainen, "Confidence as Bayesian probability: From neural origins to behavior," *Neuron*, vol. 88, no. 1, pp. 78–92, 2015.

[3] A. Pouget, J. Drugowitsch, and A. Kepecs, "Confidence and certainty: Distinct probabilistic quantities for different goals," *Nat. Neurosci.*, vol. 19, no. 3, pp. 366–374, 2016.

[4] M. O. Ernst and M. S. Banks, "Humans integrate visual and haptic information in a statistically optimal fashion," *Nature*, vol. 415, no. 6870, pp. 429–433, 2002.

[5] K. P. Körding and D. M. Wolpert, "Bayesian integration in sensorimotor learning," *Nature*, vol. 427, no. 6971, pp. 244–247, 2004.

[6] D. Kersten, P. Mamassian, and A. Yuille, "Object perception as Bayesian inference," *Annu. Rev. Psychol.*, vol. 55, pp. 271–304, Feb. 2004.

[7] Z. Shi, R. M. Church, and W. H. Meck, "Bayesian optimization of time perception," *Trends Cogn. Sci.*, vol. 17, no. 11, pp. 556–564, 2013.

[8] C. Chandrasekaran, "Computational principles and models of multisensory integration," *Current Opin. Neurobiol.*, vol. 43, pp. 25–34, Apr. 2017.

[9] D. Alais and D. Burr, "Cue combination within a Bayesian framework," in *Multisensory Processes*. Cham, Switzerland: Springer, 2019, pp. 9–31.

[10] S. Goldwater, T. L. Griffiths, and M. Johnson, "A Bayesian framework for word segmentation: Exploring the effects of context," *Cognition*, vol. 112, no. 1, pp. 21–54, 2009.

[11] N. Chater, J. B. Tenenbaum, and A. Yuille, "Probabilistic models of cognition: Where next?" *Trends Cogn. Sci.*, vol. 10, no. 7, pp. 292–293, 2006.

[12] J. Austerweil, S. Gershman, J. Tenenbaum, and T. Griffiths, "Structure and flexibility in Bayesian models of cognition," in *Oxford Handbook of Computational and Mathematical Psychology*. Oxford, U.K.: Oxford Univ. Press, 2015.

[13] K. P. Körding and D. M. Wolpert, "Bayesian decision theory in sensorimotor control," *Trends Cogn. Sci.*, vol. 10, no. 7, pp. 319–326, 2006.

[14] P. M. Bays and D. M. Wolpert, "Computational principles of sensorimotor control that minimize uncertainty and variability," *J. Physiol.*, vol. 578, no. 2, pp. 387–396, 2007.

[15] J. M. Beck *et al.*, "Probabilistic population codes for Bayesian decision making," *Neuron*, vol. 60, no. 6, pp. 1142–1152, 2008.

[16] D. Lee and H. Seo, "Neural basis of strategic decision making," *Trends Neurosci.*, vol. 39, no. 1, pp. 40–48, 2016.

[17] R. M. Haefner, P. Berkes, and J. Fiser, "Perceptual decision-making as probabilistic inference by neural sampling," *Neuron*, vol. 90, no. 3, pp. 649–660, 2016.

[18] Z. Yu, F. Chen, and F. Deng, "Unification of MAP estimation and marginal inference in recurrent neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 11, pp. 5761–5766, Nov. 2018.

[19] S. B. Furber, F. Galluppi, S. Temple, and L. A. Plana, "The Spinnaker project," *Proc. IEEE*, vol. 102, no. 5, pp. 652–665, May 2014.

[20] P. A. Merolla *et al.*, "A million spiking-neuron integrated circuit with a scalable communication network and interface," *Science*, vol. 345, no. 6197, pp. 668–673, 2014.

[21] B. V. Benjamin *et al.*, "Neurogrid: A mixed-analog-digital multichip system for large-scale neural simulations," *Proc. IEEE*, vol. 102, no. 5, pp. 699–716, May 2014.

[22] L. P. Shi *et al.*, "Development of a neuromorphic computing system," in *Proc. IEEE Int. Electron Devices Meeting*, 2015, pp. 72–75.

[23] J. C. Shen *et al.*, "Darwin: A neuromorphic hardware co-processor based on spiking neural networks," *Sci. China Inf. Sci.*, vol. 59, no. 2, pp. 1–5, 2016.

[24] J. Fiser, P. Berkes, G. Orbán, and M. Lengyel, "Statistically optimal perception and learning: From behavior to neural representations," *Trends Cogn. Sci.*, vol. 14, no. 3, pp. 119–130, 2010.

[25] W. J. Ma, J. M. Beck, and A. Pouget, "Spiking networks for Bayesian inference and choice," *Current Opin. Neurobiol.*, vol. 18, no. 2, pp. 217–222, 2008.

[26] T. J. Anastasio, P. E. Patton, and K. Belkacem-Boussaid, "Using Bayes' rule to model multisensory enhancement in the superior colliculus," *Neural Comput.*, vol. 12, no. 5, pp. 1165–1187, 2000.

[27] Z. Yu, F. Chen, and J. Dong, "Neural network implementation of inference on binary Markov random fields with probability coding," *Appl. Math. Comput.*, vol. 301, pp. 193–200, May 2017.

[28] R. P. N. Rao, "Bayesian computation in recurrent neural circuits," *Neural Comput.*, vol. 16, no. 1, pp. 1–38, 2004.

[29] R. P. N. Rao, "Hierarchical Bayesian inference in networks of spiking neurons," in *Proc. Adv. Neural Inf. Process. Syst.*, 2004, pp. 1113–1120.

[30] J. M. Beck and A. Pouget, "Exact inferences in a neural implementation of a hidden Markov model," *Neural Comput.*, vol. 19, no. 5, pp. 1344–1361, 2007.

[31] J. Y. Angela and P. Dayan, "Inference, attention, and decision in a Bayesian neural architecture," in *Proc. Adv. Neural Inf. Process. Syst.*, 2004, pp. 1577–1584.

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

[33] W. J. Ma and M. Jazayeri, "Neural coding of uncertainty and probability," *Annu. Rev. Neurosci.*, vol. 37, pp. 205–220, Jul. 2014.

[34] J. M. Beck, P. E. Latham, and A. Pouget, "Marginalization in neural circuits with divisive normalization," *J. Neurosci.*, vol. 31, no. 43, pp. 15310–15319, 2011.

[35] W. J. Ma and M. Rahmati, "Towards a neural implementation of causal inference in cue combination," *Multisensory Res.*, vol. 26, nos. 1–2, pp. 159–176, 2013.

[36] L. Buesing, J. Bill, B. Nessler, and W. Maass, "Neural dynamics as sampling: A model for stochastic computation in recurrent networks of spiking neurons," *PLoS Comput. Biol.*, vol. 7, no. 11, 2011, Art. no. e1002211.

[37] D. Pecevski, L. Buesing, and W. Maass, "Probabilistic inference in general graphical models through sampling in stochastic networks of spiking neurons," *PLoS Comput. Biol.*, vol. 7, no. 12, 2011, Art. no. e1002294.

[38] L. Shi and T. L. Griffiths, "Neural implementation of hierarchical Bayesian inference by importance sampling," in *Proc. Adv. Neural Inf. Process. Syst.*, 2009, pp. 1669–1677.

[39] S. Shipp, R. A. Adams, and K. J. Friston, "Reflections on agranular architecture: Predictive coding in the motor cortex," *Trends Neurosci.*, vol. 36, no. 12, pp. 706–716, 2013.

[40] Z. Yu, T. Huang, and J. K. Liu, "Implementation of Bayesian inference in distributed neural networks," in *Proc. 26th Euromicro Int. Conf. Parallel Distrib. Netw. Process. (PDP)*, 2018, pp. 666–673.

[41] D. Koller and N. Friedman, *Probabilistic Graphical Models: Principles and Techniques*. Cambridge, MA, USA: MIT Press, 2009.

[42] J. S. Yedidia, W. T. Freeman, and Y. Weiss, "Constructing free-energy approximations and generalized belief propagation algorithms," *IEEE Trans. Inf. Theory*, vol. 51, no. 7, pp. 2282–2312, Jul. 2005.

[43] C. Andrieu, N. De Freitas, A. Doucet, and M. I. Jordan, "An introduction to MCMC for machine learning," *Mach. Learn.*, vol. 50, nos. 1–2, pp. 5–43, 2003.

[44] K. Nagata and S. Watanabe, "Exchange Monte Carlo sampling from Bayesian posterior for singular learning machines," *IEEE Trans. Neural Netw.*, vol. 19, no. 7, pp. 1253–1266, Jul. 2008.

[45] D. George and J. Hawkins, "Towards a mathematical theory of cortical micro-circuits," *PLoS Comput. Biol.*, vol. 5, no. 10, 2009, Art. no. e1000532.

[46] A. Steimer, W. Maass, and R. Douglas, "Belief propagation in networks of spiking neurons," *Neural Comput.*, vol. 21, no. 9, pp. 2502–2523, 2009.

[47] S. Litvak and S. Ullman, "Cortical circuitry implementing graphical models," *Neural Comput.*, vol. 21, no. 11, pp. 3010–3056, 2009.

[48] J. Cheng and M. J. Druzdzel, "AIS-BN: An adaptive importance sampling algorithm for evidential reasoning in large Bayesian networks," *J. Artif. Intell. Res.*, vol. 13, no. 1, pp. 155–188, 2000.

[49] Y. Bengio and J.-S. Senécal, "Adaptive importance sampling to accelerate training of a neural probabilistic language model," *IEEE Trans. Neural Netw.*, vol. 19, no. 4, pp. 713–722, Apr. 2008.

[50] M. Kouh and T. Poggio, "A canonical neural circuit for cortical nonlinear operations," *Neural Comput.*, vol. 20, no. 6, pp. 1427–1451, 2008.

[51] S. J. Mitchell and R. A. Silver, "Shunting inhibition modulates neuronal gain during synaptic excitation," *Neuron*, vol. 38, no. 3, pp. 433–445, 2003.

[52] J. S. Rothman, L. Cathala, V. Steuber, and R. A. Silver, "Synaptic depression enables neuronal gain control," *Nature*, vol. 457, no. 7232, pp. 1015–1018, 2009.

[53] O. Braganza and H. Beck, "The circuit motif as a conceptual tool for multilevel neuroscience," *Trends Neurosci.*, vol. 41, no. 3, p. 128, 2018.

[54] R. Douglas and K. Martin, "Neuronal circuits of the neocortex," *Annu. Rev. Neurosci.*, vol. 27, pp. 419–451, Jul. 2004.

[55] D. R. Wozny, U. R. Beierholm, and L. Shams, "Human trimodal perception follows optimal statistical inference," *J. Vis.*, vol. 8, no. 3, p. 24, 2008.

[56] A. Yuille and R. Mottaghi, "Complexity of representation and inference in compositional models with part sharing," *J. Mach. Learn. Res.*, vol. 17, no. 1, pp. 292–319, 2016.

[57] Y. Jin and S. Geman, "Context and hierarchy in a probabilistic image model," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2, 2006, pp. 2145–2152.

[58] P. Luo, L. Lin, and X. Liu, "Learning compositional shape models of multiple distance metrics by information projection," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 7, pp. 1417–1428, Jul. 2016.

[59] J. A. M. Lorteije, A. Zylberberg, B. G. Ouellette, C. I. De Zeeuw, M. Sigman, and P. R. Roelfsema, "The formation of hierarchical decisions in the visual cortex," *Neuron*, vol. 87, no. 6, pp. 1344–1356, 2015.

[60] M. Zhang, H. Qu, A. Belatreche, and X. Xie, "EMPD: An efficient membrane potential driven supervised learning algorithm for spiking neurons," *IEEE Trans. Cogn. Develop. Syst.*, vol. 10, no. 2, pp. 151–162, Jun. 2018.

[61] T. Masquelier and S. J. Thorpe, "Unsupervised learning of visual features through spike timing dependent plasticity," *PLoS Comput. Biol.*, vol. 3, no. 2, p. e31, 2007.

[62] F. Ponulak and A. Kasiński, "Supervised learning in spiking neural networks with ReSuMe: Sequence learning, classification, and spike shifting," *Neural Comput.*, vol. 22, no. 2, pp. 467–510, 2010.

[63] N. Frémaux and W. Gerstner, "Neuromodulated spike-timing-dependent plasticity, and theory of three-factor learning rules," *Front. Neural Circuits*, vol. 9, p. 85, Jan. 2016.

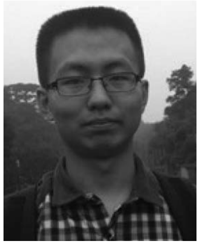
[64] S. Guo, Z. Yu, D. Fei, X. Hu, and C. Feng, "Hierarchical Bayesian inference and learning in spiking neural networks," *IEEE Trans. Cybern.*, vol. 49, no. 1, pp. 133–145, Jan. 2019.

[65] D. Pecevski and W. Maass, "Learning probabilistic inference through spike-timing-dependent plasticity," *eNeuro*, vol. 3, no. 2, p. 48, 2016.

[66] S. Song, K. D. Miller, and L. F. Abbott, "Competitive Hebbian learning through spike-timing-dependent synaptic plasticity," *Nat. Neurosci.*, vol. 3, no. 9, pp. 919–926, 2000.

[67] G. Q. Bi and M. M. Poo, "Synaptic modifications in cultured hippocampal neurons: Dependence on spike timing, synaptic strength, and postsynaptic cell type," *J. Neurosci.*, vol. 18, no. 24, pp. 10464–10472, 1998.

[68] Z. Yu, F. Chen, J. Dong, and Q. Dai, "Sampling-based causal inference in cue combination and its neural implementation," *Neurocomputing*, vol. 175, no. 1, pp. 155–165, 2016.



Zhaofei Yu (M'19) received the B.S. degree from the Hong Shen Honors School, College of Optoelectronic Engineering, Chongqing University, Chongqing, China, in 2012, and the Ph.D. degree from the Automation Department, Tsinghua University, Beijing, China, in 2017.

He is currently a Post-Doctoral Fellow with the National Engineering Laboratory for Video Technology, Department of Computer Science and Technology, Peking University, Beijing. His current research interests include brain-inspired computing and computational neuroscience.

Feng Chen (M'06) received the B.S. and M.S. degrees in automation from Saint-Petersburg Polytechnic University, Saint Petersburg, Russia, in 1994 and 1996, respectively, and the Ph.D. degree from the Automation Department, Tsinghua University, Beijing, China, in 2000.

He is currently a Professor with Tsinghua University. His current research interests include computer vision, brain-inspired computing, and inference in graphical models.

Jian K. Liu received the Ph.D. degree in mathematics from the University of California at Los Angeles, Los Angeles, CA, USA, in 2009.

He is currently a Lecturer with the Centre for Systems Neuroscience, University of Leicester, Leicester, U.K. His current research interests include computational neuroscience and brain-like computation.